LINUX FORMAT

К вашим услугам

Звезда этого номера – SimplyMepis – идеальный выбор для новичка. Поэтому мы спросили Команду LXF: **какой ценый совет вы могли бы дать человеку, начинающему знакомство с Linux?**



Пол Хадсон
Каждый раз, когда вы запускаете KDE, Бог убивает котенка. Бога ради, подумайте о котятах!



Грэм Моррисон
Важна не цель – важен процесс ее достижения.



Эфрейн Эрнандес-Мендоза
Не орите на неработающий компьютер и не кляните его – Linux не поддерживает речевое управление.



Майк Сондерс
Если с наскоку не получится – попробуйте другой дистрибутив. Особенно Ubuntu.



Ребекка Смолли
Запаситесь подставкой под руки, лампой и пачкой комиксов – я имею в виду книжки O'Reilly.



Эндрю Грегори
Каждый раз, когда вы устанавливаете Linux, помните, что лицензия на Windows стоит 200 долларов.



Дэвид Картрайт
Не надейтесь, что все заработает через минуту. Копирование файлов происходит не мгновенно, установкам требуется время, чтобы устояться.



Энди Ченнел
Установите себе VMware Player. Виртуализация – это круто!



Нейл Ботвик
Помните о своих лучших друзьях – Google и форумах LXF.



Ричард Коббет
Каждый, кто высмеял вас в ответ на просьбу о помощи – козел. Не обращайте внимания. Продолжайте попытки.



Энди Хадсон
Пейте побольше кофе, чтобы не заснуть; делайте побольше ошибок, чтобы не заскучать.



Ладислав Боднар
Удалите Windows с жесткого диска. Это может показаться страшным, но через три недели вы о ней даже не пожалеете.

О свободе и свободном ПО

Недавно на форуме LXF проскочила занятная шутка: скоро-де наш журнал, в соответствии с идеологией Linux, будет распространяться в виде отдельных листов, а читатели сами, при помощи клея и ножниц, буду собирать нужное им издание (рискну предположить, что эту колонку при сборке своей собственной версии LXF многие «сотрут», так что читайте пока можете). Забавно? Забавно! А теперь представьте себе очкарика из пыльных недров типографии, который заходит на этот форум и брызжа слюной напополам с ядом, начинает убеждать всех, что именно это и есть правильный путь чтения журналов и он-де занимается этим всю сознательную жизнь, а остальные и звания читателя-то не заслуживают. Не думаю, что это кому-нибудь понравится, а вот порассуждать на тему: «Мы на Gentoo – крутота, вы с Fedora – беднота» (фраза инвариантна относительно перестановки названий дистрибутивов) любит едва ли не каждый второй. При этом бытует мнение, что правильный пользователь Linux – это тот, который пересобирает ядро, да еще и из make config, и на закуску – с парой патчей от хакера Васи. Оглянитесь вокруг: далеко не всем интересно, как устроен дистрибутив изнутри. Некоторым гораздо важнее быстро оформить документ или написать письмо. Так что же, Gentoo, Slackware, Arch Linux и прочие - «неправильные» дистрибутивы? Конечно же, нет. Просто они преследуют другие цели.

Нельзя делить программное обеспечение на правильное и неправильное. Свободное ПО дает нам возможность создать систему, которая будет идеально подходить для целей конкретной, быть может – немногочисленной группы единомышленников. Убеждать всех и каждого, что единственно правильный дистрибутив – это «X» – антисвободно.



Мы тоже не собираемся навязывать вам свой выбор. В этом месяце на диске Linux Format вы найдете сразу три операционных системы: Mepis для «пользователей», Gentoo – для «хакеров» и OpenSolaris – для хакеров постарше. Наслаждайтесь!

С уважением,
СИНИЦЫН ВАЛЕНТИН
главный редактор LinuxFormat в России

КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации:
letters@linuxformat.ru
Подписка и предыдущие номера:
subscribe@linuxformat.ru
Техническая поддержка:
answers@linuxformat.ru
Проблемы с дисками:
disks@linuxformat.ru
Общие вопросы: **info@linuxformat.ru**
Web-сайт: **www.linuxformat.ru**

Адрес редакции: Россия,
Санкт-Петербург, ул. Гончарная, 23, офис 54
Телефон редакции: (812) 717-00-37
Дополнительная информация на стр.128

МИССИЯ ЖУРНАЛА

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Corporate Patron Free Software Foundation

Содержание

LINUX FORMAT

Добро пожаловать в *LinuxFormat* – ваш гид в мире Linux!

LXF5(79) МАЙ 2006



» ЧИТАЙТЕ В ЭТОМ ВЫПУСКЕ



МЫСЛИ ВСЛУХ

«Я понял, что **MEPIS** имеет потенциал. Его можно расшифровать как **Моя Экстраординарная Персональная Информационная система**.»



«Сейчас я займусь **созданием подкаста**, в основу которого ляжет поэма Роберта Брауна «**Дудочник в пестром костюме из Гамелина**.»

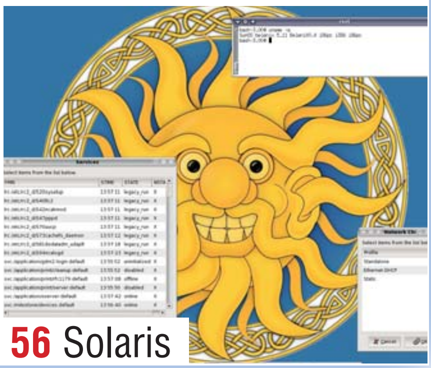




12 eth0 в розетке



40 KDE Solid



56 Solaris

На диске

Целый DVD интересных программ 120



» DVD

SIMPLYMEPIS Вставьте в DVD, перезагрузитесь, готово! Самый простой Linux

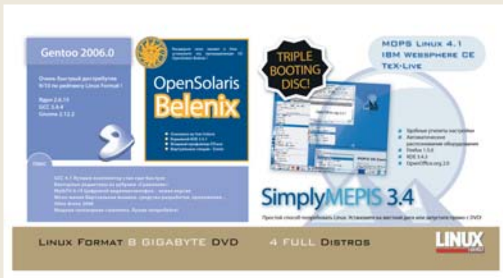
BELENIX Дистрибутив OpenSolaris

GENTOO Супер-настраиваемый дистрибутив, который не остановится ни перед чем в погоне за скоростью.

MONO МАНИЯ Все, что нужно для создания и запуска приложений .NET

GCC 4.1 Новая коллекция компиляторов GNU

Вырежьте свою обложку для DVD на стр. **125-126**



Перед использованием диска ознакомьтесь с инструкцией на стр. 125



Главные новости

• KOffice 1.5 • Portland • Red Hat приобретает JBoss • Конец Fedora Foundation

KOffice 1.5

Интегрированный офисный пакет KDE готовится встать на одну ступень с OpenOffice.org и Microsoft Office

11 апреля команда разработчиков *KOffice* (www.KOffice.org) представила новую крупную версию этого интегрированного офисного пакета под номером 1.5. Как утверждают сами создатели, *KOffice 1.5* повзрослел и готов составить конкуренцию *OpenOffice.org* и Microsoft Office. Изюминкой этого релиза стала поддержка формата *OASIS OpenDocument* (ODF). Таким образом, *KOffice* 1.5 стал вторым (после OOo 2.0) пакетом, способным читать и записывать ODF-файлы. *OpenDocument* является родным форматом для *KWord* (.odt), *KSpread* (.ods) и *KPresenter* (.odp); *KChart* и *KFormula* также поддерживают его, но не в полной мере. Как отмечают разработчики, *KOffice 1.5* может испытывать проблемы с переносимостью ODF-документов. Эту неприятность планируется устранить в ходе выпуска ближайших исправлений – 1.5.x

Вторым значимым нововведением является *Kexi 1.0* – настольная база данных, являющаяся аналогом Microsoft Access (базы данных которого можно импортировать) и *OpenOffice.org Base*. Kexi может работать как с файловой БД, так и с полноценным сервером, например, *MySQL* или *PostgreSQL*. Примечательно, что Kexi является первой коммерчески поддерживаемой частью *KOffice*, портированной к тому же в Microsoft Windows.

Ну и наконец, нельзя не упомянуть *Kross* – среду (framework) для разработки сценариев автоматизации *KOffice*, выполняющую те же функции, что и *VBA* в *MS Office*. Kross не зависит от языка программирования (заявлены *Python*, *Ruby*, *JavaScript* и даже сценарии оболочки, посредством IPC), однако, несмотря на общее предназначение, не нацелен на совместимость с *VBA*. Максимум, что может обещать команда *KOffice* – постараться сделать объектную модель *KSpread* ближе к Excel, что упростит процесс переноса макросов.

KOffice традиционно оставался в тени гигантов, однако, учитывая темпы разработки, ситуация может измениться уже в ближайшие год-два. Большие надежды возлагаются на *KOffice* 2.0, который должен выйти примерно в то же время, что и KDE4.

• Читайте обзор *KOffice* в следующем номере *Linux Format*!

Intel и Red Hat ускоряют развертывание Linux

Компании приступили к реализации глобальной программы, которая поможет планировать и оптимизировать внедрение решений на базе Linux

«Программа *Red Hat* and Intel Solution Acceleration Program позволит клиентам своевременно получать критически важную информацию, инструменты и услуги поддержки, необходимые для создания и оптимизации передовых решений на базе Linux и платформ Intel, – сказал Йон Борк (Jon Bork), руководитель программы Open Source Program корпорации Intel. – Эта программа поможет компаниям быстро и эффективно использовать преимущества новых платформ и технологий Intel, как только они будут появляться на рынке».

Программа *Red Hat* and Intel Solution Acceleration Program будет реализована как с помощью онлайновых средств, так и специализированных центров, расположенных в городах Маклин (штат Вирджиния), Мумбай (Индия) и Мюнхен (Германия), а также в 14 других городах по всему миру. Центры будут оснащены серверами на базе процессоров Intel Itanium 2 и Intel Xeon, корпоративными настольными ПК на базе процессоров Intel Pentium 4, ноутбуками на базе технологии Intel Centrino для мобильных ПК и системами хранения данных на базе компонентов Intel, в том числе платформами, пока не запущенными в серийное производство. Все центры будут организованы на объектах компании *Red Hat* и будут доступны экспертам по сервисам и решениям из компаний Intel и *Red Hat*, а также их клиентам и партнерам, участвующим в развертывании решений. В число сервисов, доступных через данные центры и в онлайновом режиме, войдут: тренинг и распространение знаний, поддержка опытно-экспериментальных проектов, создание продукции для реселлеров и вертикальных сегментов рынка, а также передовые проекты обмена знаниями. Например, в центрах программы будут реализованы пилотные проекты по оценке эффективности технологии Intel Virtualization Technology в средах Fedora Core 5 и *Red Hat* Enterprise Linux 5 с использованием гипервизора *Xen*.

К реализации программы Solution Acceleration Program компании *Red Hat* и Intel приступят в апреле месяце.

Portland делает первые шаги

Хит сезона – межпроцедурная оптимизация

Первая публичная версия «клея», связывающего KDE и Gnome, была обнародована на выставке LinuxWorld в Бостоне

Проект *Portland* (http://portland.freedesktop.org), цель которого – объединить KDE и Gnome (или, более точно, предоставить разработчикам набор API и утилит, который позволит создавать приложения с помощью любого инструментария, не задаваясь вопросами интегрированности с рабочим столом пользователя), представил на суд публики первые наработки. Предварительные версии утилит командной строки и библиотек – *xdg-utils Technology Preview* и *DAPI Technology Preview* – уже можно загрузить с сайта проекта. *Xdg-utils* – «прослойка» из утилит командной строки, выполняющих типичные действия (копирование файла, запуск программы от имени другого пользователя, открытие почтового клиента и web-браузера и т.д.) способом, предусмотренным разработчиками графической среды/дистрибутива. При этом простейшие задачи интеграции существенно упростятся: если все сторонние производители будут использовать xdg-utils, создателям рабочих столов останется лишь убедиться в корректной работе этих утилит. DAPI выполняет аналогичные функции, но уже на уровне исходного кода. Ее составные части – это библиотека *libdapi.la*, с которой компонуется приложение третьей фирмы и демон, специфичный для данной оконной среды. Поскольку все демоны используют одинаковые интерфейсы API, программа может вызывать их функции (посредством библиотеки), не заботясь о том, как именно происходит вызов.

Portland динамично развивается и, если все пойдет, как предполагается, вы сможете увидеть бета- или даже финальную версию пакета к тому моменту, когда будете читать эти строки.

Red Hat приобретает JBoss

Покупка обойдется Red Hat в 350 млн. долларов

10 апреля 2006 года компания *Red Hat* объявила о своем намерении приобрести фирму *JBoss*, производителя одноименного открытого сервера приложений J2EE. Все необходимые договоренности уже достигнуты, а сама сделка должна завершиться в мае. Покупка *JBoss* обошлась Red Hat в 350 миллионов долларов, 60 процентов которых составляют непривилегированные акции компании, однако, при достижении определенных показателей, эта сумма может быть увеличена до 420 млн. Бизнес *JBoss*, как и *Red Hat*, построен на модели подписки – доставки определенных услуг (в первую очередь, поддержки) через сеть. Эта фирма может предложить то, чего так долго не хватало *Red Hat* – промежуточное ПО (middleware), необходимое для развертывания на базе Red Hat Enterprise Linux J2EE-приложений уровня предприятия. «Red Hat и JBoss уверены, что открытая модель разработки будет продолжать менять экономику мира IT в пользу потребителя. Мы верим в инновационный потенциал, возникающий после отказа от проприетарной разработки»,-сказал исполнительный директор Red Hat Мэттью Сцулик (Matthew Szulik).

Конец Fedora Foundation

Участники проекта решили «пойти другим путем»

В июне прошлого года компания *Red Hat* объявила о своем намерении передать контроль над проектом *Fedora* фонду Fedora Foundation. Однако, попытки по созданию этого Фонда не увенчались успехом. В итоге, построение Fedora Foundation было решено заменить реструктуризацией внутри самого проекта.

Фонд *Fedora* – детище Марка Веббинка (Mark Webbink), специалиста по интеллектуальной собственности в *Red Hat*, Inc. и задумывался он с основной целью – обеспечить репозитарий патентов, защищающих открытое сообщество. Однако, вскоре после анонса Фонд оброс множеством дополнительных направлений и стал чересчур неповоротливым, поэтому в начале апреля разработчики предложили новую модель управления проектом. Было решено допустить к рычагам власти «стихийных лидеров», которые определились естественным путем в ходе развития проекта. В настоящий момент в состав «совета директоров» *Fedora* входит 9 человек, 5 из которых являются сотрудниками *Red Hat*. Председателем Совета также является человек, назначенный компанией. Сейчас эту должность занимает Макс Спевак (Max Spevack).

Алексей Федорчук
Переехал на Ubuntu с Arch Linux.



КОММЕНТАРИИ

Kubuntu в роли пасынка?

Как известно, проект Ubuntu и его «дочерние предприятия» – Kubuntu и Edubuntu, – финансируются Марком Шаттлвортом. Однако он не спонсирует разработчиков напрямую: для этого существует специальная компания Canonical (http://canonical.com/), зарегистрированная на острове Мэн (Ирландское море), в задачу которой и входит справедливое распределение средств, выделяемых космонавтом-линуксоидом. Однако именно справедливость распределения и была поставлена под сомнение разработчиками Kubuntu.

В частности, и потому, что множественное число в отношении последних – некоторое преувеличение: кроме дюжины энтузиастов, по штату этим делом занимается один-единственный человек, Джонатан Риддел из Эдинбурга. Именно ему пользователи Kubuntu обязаны рекордными по срокам сборками новейших (и тестируемых) версий KDE. Недавнее интервью с ним можно прочитать здесь: http://behindubuntu.org/interviews/JonathanRiddell/ (правда, в нем Риддел на жизнь не жалуется).

Не включена Kubuntu и в программу бесплатного распространения по всему миру, как это делается для «головного» дистрибтива. И это при том, что, по слухам, на машине самого Марка стоит именно Kubuntu...

КОРОТКОЙ СТРОКОЙ

- Началась подготовка к выпуску рабочей среды **Xfce 4.4**, включающей долгожданный файловый менеджер Thunar.
- Проект **Arch64**, разрабатывающий 64-битную версию дистрибутива Arch Linux, наконец-то представил публике установочный **ISO-образ**. По ассортименту пакетов Arch64 практически не уступает своему «родителю».
- Фонд **Mozilla Foundation** обновил браузеры семейства Firefox: 1.5.0.2 и 1.0.8.
- Вышла в свет бета-версия **Mono 1.2**, включающая в себя полноценную реализацию Windows.Forms. Это упростит перенос существующих .NET-приложений в Linux.

WINE@ETERSOFT 1.0 LOCAL

Компания Этерсофт совместно с компанией Линуксцентр представляет продукт WINE@Etersoft 1.0 Local, предназначенный для запуска Windows-приложений в свободных операционных системах на платформе x86. В новой версии учтено большое число замечаний и предложений, поступивших от пользователей версии 0.9.

WINE@Etersoft позволяет использовать такие российские программы как 1С:Предприятие 7.7, 1С: Бухгалтерия 6.0, Консультант Плюс, Инфо-Бухгалтер, Кодекс, Референт, ДубльГИС, программы подготовки обязательной отчётности, а также MS Office 97 и некоторые другие в операционных системах.



* ALT Linux
* ASP Linux
* Debian, Knoppix
* Fedora Core
* FreeBSD
* LinuxXP
* Mandriva
* MOPSLinux
* Novel Linux Desktop
* Scientific Linux, Cent OS
* Slackware
* Open SuSe
* Ubuntu

Версия WINE@Etersoft Local ориентирована на индивидуальных пользователей, которым требуется работать с win-приложениями на платформе Linux/Unix. В стоимость продукта включён пакет сервисной поддержки в течение 30 дней

Подведение итогов конкурса «SOFTOOL: Дебют года-2005»

24 апреля 2006 года в Москве, в помещении Российской Академии Наук (РАН), прошла торжественная церемония награждения победителей конкурса «SOFTOOL: Дебют года-2005». Премия «Дебют года» учреждена как Российская национальная некоммерческая негосударственная премия и вручаются продукту-победителю Конкурса «SOFTOOL: Дебют года» по результатам голосования экспертной комиссии конкурса. Учредителями «SOFTOOL: Дебют года-2005» выступили Федеральное Агентство по информационным технологиям, Отделение Информационных Технологий и Вычислительных Систем Российской Академии Наук, издательство «Открытые системы» и компания «ИТ-ЭКСПО».

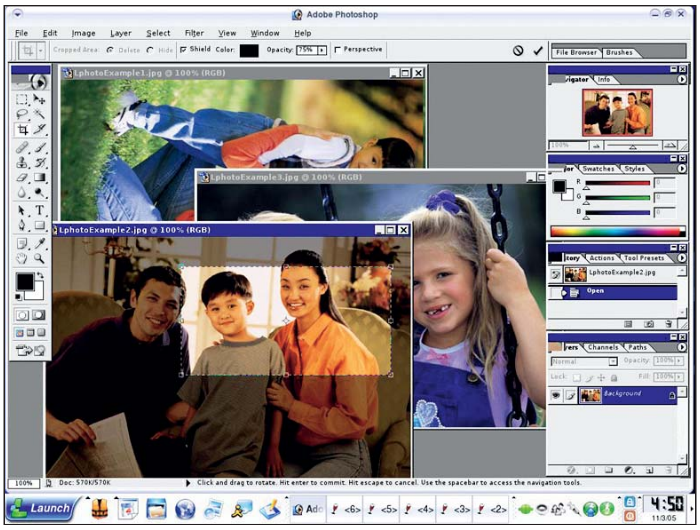
На конкурс было представлено 64 работы, но победителями стали лишь 14 из них. Нам особенно приятно отметить, что в число лауреатов вошел отечественный дистрибутив Linux – MOPSLinux 3.1, развиваемый совместными усилиями компаний НПО «СЕТЬ» (www.rpunet.ru) и LinuxCenter.Ru (www.linuxcenter.ru). MOPSLinux – полностью русифицированная операционная система, в основе которой лежит Slackware GNU/Linux. Разработчики MOPSLinux расширили ассортимент пакетов родительской системы и обеспечили продукт технической поддержкой. Изданием и распространением MOPSLinux занимается компания LinuxCenter.Ru.



Хорошие новости для тех, кто умеет ждать

CrossOver приносит Windows-приложения в дома пользователей Linspire

Пользователи Linspire дождались приложений Windows на своих компьютерах благодаря включению специальной версии CrossOver Office в Click-N-Run Warehouse своей ОС. Кевин Камони (Kevin Carmony), высокопоставленный представитель Linspire отметил, что многие представители бизнеса, выбирая Linux в качестве настольного решения были вынуждены мириться с потерей нескольких жизненно важных Windows-приложений. «CrossOver Office вычеркивает любые сомнения из формулы выбора. Теперь можно использовать нужные программы, не переплачивая за операционную систему,» – подчеркнул Камони.



Пакет с CrossOver был тщательно адаптирован под Linspire, так что приложения оказываются в нужных частях меню, да и ошибки, которые препятствовали эффективной работе пакета в составе дистрибутива, повыведены.

CrossOver значительно повышает совместимость Linspire с Windows-продуктами, заявленную изначально, но до сих пор обеспечивавшуюся только через Wine: отныне доступны Photoshop и Dreamweaver; Quicken и Quickbooks от Intuit; Lotus Notes – и все благодаря CrossOver Office.

СОБЫТИЯ: ЧТО БУДЕТ

Санкт-Петербургская конференция Интернет и Реклама

31 мая 2006 года в отеле Ольгино в рамках Санкт-Петербургской конференции Интернет и Реклама пройдет Всероссийский Интернет Марафон.

Аудитория Рунета стремительно растет, а вместе с ней растут и обороты Интернет-индустрии. Сегодня Интернетом пользуется более 21% россиян, объем регистрации доменов растет на 46% в год, а по скорости развития домен .RU занимает второе место в мире после Китая.

В Северо-западном Регионе Интернет-индустрия развивается очень активно. В Санкт-Петербурге уже более 1 500 000 пользователей сети Интернет, и все новые и новые компании задаются вопросом о том, как правильно организовать свой бизнес в Сети. Однако новичкам довольно трудно сориентироваться во всем многообразии инструментов и технологий и выбрать оптимальную стратегию развития Интернет-проекта. Поэтому компании, давно и успешно работающие на рынке Санкт-Петербурга, решили организовать конференцию, цель которой - привлечение внимания компаний Северо-Западного округа к современным бизнес-возможностям Интернета и демонстрация решений актуальных бизнес-задач на практике.

Это первая в Северо-Западном Регионе конференция подобного рода. Организаторами стали компании: SEO TEAM, Битрикс , AdMedia , Мезон.Ру , TRINET и Региональный общественный центр Интернет-технологий (РОЦИТ)

Партнеры конференции: Hewlett-Packard, R-Style, Webalta, Liveinternet, 3DNews, iXBT.com, Subscribe.Ru, NNM.ru, AdLife.spb.ru, ITNews.spb.ru, Росбалт, Издательство Открытые Системы, PC-Mobile, T3, Деловой Петербург, НП ПРИОР СЗ, Linux Format, Free-lance.ru

Участие для слушателей и докладчиков бесплатное. Зарегистрироваться можно, заполнив форму на сайте конференции www.spkir.ru. Действие развернется на территории отеля «Ольгино», расположенного на берегу Финского залива в 20 минутах езды от города. Организаторы обеспечивают бесплатный транспорт для участников конференции.

На конференции будут охвачены все этапы создания и продвижения Интернет-проекта. В рамках мероприятия выступят ведущие специалисты отрасли с докладами по следующим направлениям:

- **Аудитория Рунета**
Анализ перспектив развития

- **Технологии Рунета**
Системы управления Интернет-проектом;
Информационная безопасность;
Бизнес-процессы в Интернете

- **Продвижение и реклама в Рунете**
Поисковая оптимизация;
методики проведения рекламных кампаний и новые технологии

- **Мастер-классы**
Как правильно собрать и настроить веб-сервер;
Как управлять сайтом;
Как оптимизировать сайт под поисковые системы;
Как провести рекламную кампанию;
Как проанализировать эффективность работы сайта

По просьбам представителей крупных Интернет-проектов будут подробно рассмотрены вопросы сборки и настройки веб-серверов на базе ОС Linux, обеспечения информационной безопасности и стабильной работы сайта под высокой нагрузкой. Так же в рамках конференции специалисты компании R-Style проведут эксклюзивная презентация нового (пятого) поколения серверов Proliant от Hewlett Packard, официально не объявленных на рынке (только для подписавших соглашение о неразглашении информации, полученной на презентации).

В рамках Санкт-Петербургской конференции Интернет и Реклама проходит Всероссийский Интернет-марафон (www.internetmarafon.ru) организованный РОЦИТ (Региональный общественный центр Интернет-технологий). Цель Интернет-марафона – популяризация Интернет-технологий в российских регионах, обмен опытом в ИТ-сфере на региональном уровне, а также обеспечение базиса для Всероссийского Фестиваля региональных Интернет-проектов «Новая реальность».

Патронат Интернет-марафона: Федеральное агентство по печати и массовым коммуникациям.

Поддержка Интернет-марафона: Министерство промышленности и энергетики РФ, Федеральное агентство по информационным технологиям, Федеральное агентство по образованию.

Получить более подробную информацию по конференции, заполнить заявку на участие можно на официальном сайте мероприятия www.spkir.ru и на сайте www.internetmarafon.ru



Reader Awards 2006

Голосование завершено – время назвать победителей.

Последние несколько месяцев читатели *Linux Format* со всего света выдвигали открытые проекты на получение престижной награды Linux Format Reader Awards 2006. Особая ценность этого титула в том, что его присуждаете только вы – пользователи Linux и разработчики.

Настало время назвать имена победителей. Ниже представлены номинанты в 14 категориях (краткое описание которых можно найти на сайте). Обратите также внимание на врезку «Как проголосовать» внизу страницы. Все, что вам нужно – назвать имена достойных, LXF позаботится об остальном.

Лучший дистрибутив
Debian
Gentoo
Mandriva
SUSE /OpenSUSE
Ubuntu

Лучшее настольное приложение
Amarok
Firefox
Beagle
KDE
OpenOffice.org

Лучшее Интернет-ПО
Firefox
Joomla
Mambo
Mozilla Thunderbird
Opera

Лучшая игра для Linux
Freeciv
Frozen Bubble
Quake 4
SuperTux
Unreal Tournament

Открытый проект года
Firefox
Gnome
Joomla
KDE
Mambo

Лучшее бизнес-решение
Canonical
IBM
Nessus
Novell
Red Hat

Лучшее офисное ПО
AbiWord
CrossOver Office
Gnumeric
KOffice
OpenOffice.org

Лучшее серверное ПО
Apache
MySQL
PHP
Postfix
PostgreSQL

Лучший инструмент разработчика
Eclipse
GCC
KDevelop
Quanta
Vim

Лучшая поддержка оборудования
HP
IBM
Intel
Novel
Nvidia

Лучшее встроенное Linux-решение
Familiar
Nokia 770
Montavista
Motorola
Trolltech

Лучший хостинг или провайдер
DreamHost
NTL
Rochen
UKLinux.net

Лучший справочный ресурс
Gentoo forums
Google
IRC
LinuxQuestions.org
Ubuntu forums

Лучший гаджет года
GP2X
iPod
Nokia 770
PSP
Sharp Zaurus

КАК ПРОГОЛОСОВАТЬ

Хотите сказать «спасибо» за прекрасную работу на благо сообщества?

Голосование проходит в режиме он-лайн по адресу: www.linuxformat.co.uk/awards. Там вы найдете форму и пояснение к каждой из номинаций. Голосование закрывается 9 июня, а его результаты будут опубликованы в *LXF83*. Еще раз напоминаем: победители конкурса Linux Format Reader Awards определяются только вами. Это престижная награда и мы будем строго следить за ходом голосования. Вы можете выбрать номинантов в каждой из категорий только единожды – если вы передумаете, предыдущие результаты не будут учтены. Благодарим за участие! **LXF**

Обзоры >>

Новинки программного и аппаратного обеспечения с точки зрения наших экспертов.



12 MicroLink dLAN

OPEN SOURCE: РАЗРАБОТЧИКИ И СПОНСОРЫ



Алексей Федорчук
Пропустил шумиху вокруг FC5 мимо ушей.

Когда Red Hat и Novell диверсифицировали линейки своих Linux-дистрибутивов на как бы коммерческие и стопроцентно свободные ветви, они пошли разными путями. Во взаимоотношених SUSE и OpenSUSE все ясно: вторая представляет собой, по сути, бета-версию собственно коммерческого продукта, на которой обкатываются инновации – то есть это SUSE для смелых.

Взаимоотношения же Red Hat и Fedora Core эволюционировали во времени. Сначала наследница Корнея Чуковского представляла собой просто Красную Шляпу с чуть подрезанными (за счет не

«ПРОЕКТ FEDORA CORE ВОЗВРАЩАЕТСЯ ПОД КРЫЛО РОДИТЕЛЬСКОЙ КОМПАНИИ.»

вполне свободного софта) полями и примятой (из-за отсутствия техподдержки) тульей. То есть своего рода Red Hat для бедных. Затем «беднякам» была предоставлена некоторая свобода – создается Fedora Foundation, призванный управлять проектом самостоятельно. И вот теперь поступает заявление, что фонд со своей задачей не справился, и проект Fedora Core возвращается под крыло родительской компании. Правда, руководство им будет осуществляться на паритеных началах с независимыми разработчиками.

Как это скажется на пользователях? Да, скорее всего, никак. Те, кто может себе позволить, будут продолжать покупать Red Hat ради технической поддержки. А те, кто в нем не нуждаются, найдут пути для решения возникающих проблем.

ЧТО НОВЕНЬКОГО?



10 Fedora Core 5



13 MythTV 0.19

НАШ ВЕРДИКТ: ПОЯСНЕНИЕ

Все продукты оцениваются по 10-бальной шкале. 4 обычных параметра оценки: возможности, производительность, простота использования и соотношение «цена/качество», но для свободного ПО последний параметр может быть заменен на оценку документации. Независимо от набора категорий, мы всегда вычисляем общий рейтинг, подводящий итог нашим высказываниям.

Продукты, выделяющиеся из основной массы, получают престижную награду *Linux Format Top Stuff Award*. Выбираются самые-самые – и только лишь высокая оценка здесь еще ни о чем не говорит.

Для тестирования серверов используется наша собственная разработка, LFXBench 2004, состоящая из 4 основных подсистем: Multi CPU, Single CPU, RAM и Hard Disk. Оценки усредняются и дают общий рейтинг.

1 очко означает, что тестовый компьютер равен нашему эталону: Pentium 4 1,8ГГц, 512 Мб оперативной памяти и жесткий диск IDE. 2 очка означает, что он вдвое быстрее.

Все тесты выполняются под управлением Red Hat Enterprise Linux 3 AS на соответствующей платформе (x86, AMD64, Itanium). Код компилируется при помощи GCC, если не указано иное.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

MULTI CPU	6.07
SINGLE CPU	3
RAM	2.17
HARD DISK	0.46

ОБЩИЙ РЕЙТИНГ 2.93

ДИСТРИБУТИВ LINUX

Fedora Core 5

Энди Хадсон наслаждается исключительным релизом от кипучей команды Fedora.

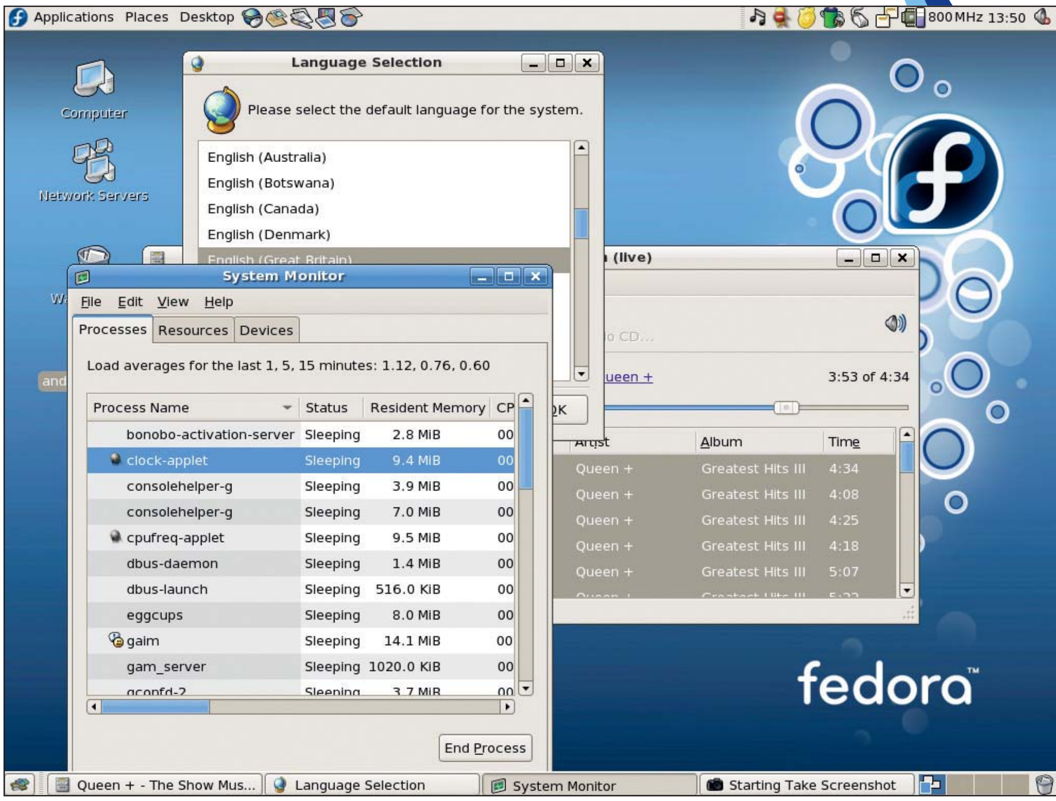
САМОЕ ГЛАВНОЕ

Community-дистрибутив с частым обновлением.
См. также OpenSUSE или Ubuntu.

- **РАЗРАБОТЧИК:** The Fedora Project
- **САЙТ:** http://fedoraproject.org
- **ЦЕНА:** Бесплатно, по лицензии GPL

Fedora Core 4 была представлена общественности девять месяцев назад. Это была первая крупная версия после выпуска Red Hat Enterprise Linux 4. В то время от нее ожидали большого скачка, однако, как помнят читатели *Linux Format*, оценка нашего обзора в *LXF69* была 4 из 10. Мы расценили ее как промежуточный релиз и с нетерпением стали ждать Fedora Core 5. В интервале между предыдущим обзором и номером, который вы держите в руках, было объявлено, что разработка FC5 потребует девяти месяцев, в нарушение ранее назначенного шестимесячного цикла. Дата выпуска еще пару раз срывалась, и в конце концов дистрибутив был опубликован почти на месяц позже графика.

Нам ужасно хотелось поглядеть, чего достигли разработчики на этот раз, причем не в последнюю очередь – насчет внешнего облика и восприятия дистрибутива, поскольку еще до выпуска Fedora 4 сообщество Fedora активно обсуждало замену обветшавшего головного убора на новый логотип. В пятой версии мы увидели окончательный результат *in situ*. И он весьма хорош. Выразительная символика: скомбинированы 'f' (от freedom – свобода, да и Fedora), знак бесконечности – бесконечный потенциал, и, наконец, речевой пузырь – популяризация. На рабочем столе он присутствует в новой теме Пузыри (Bubbles), повсеместно вытесняя красную шляпу, а цветовая схема сменилась на голубую.



Хорош логотип? О том, как сообщество помогло ему родиться – на http://fedoraproject.org/wiki/Logo.

Благодаря Cairo, Gnome выглядит лучше чем обычно. Правда, это не чистый Gnome 2.14: некоторых функций недостает, наиболее заметная из них – быстрое переключение пользователя, реализация которого признана просто уродской. (*Для более полного обзора Gnome 2.14, см. стр. 14.*)

Горнило Red Hat

По внешнему виду и ощущениям от дистрибутива, вполне простительно решить, что Fedora порвала с Red Hat, но ничто не может быть дальше от истины. Четко прослеживается, что это лабораторная разработка Red Hat на подступах к подвижной и одновременно стабильной платформе. В релиз явно вложен большой труд. К удовольствию многих, целиком реализован модульный *X.org* (7.0); *GCC* добрался до версии 4.1; и весь дистрибутив несколько раз пересобирался. Вы заметите, что значительное внимание уделено *Xen*, что и неудивительно, потому как Xen будет играть важную роль в RHEL 5. Система безопасности SELinux также подверглась изменению, в соответствии с изменением подхода Fedora к SELinux, позволяющему получить выгоду из большого сообщества. Это опять-таки важно для RHEL 5, особенно для сертификации по общим критериям безопасности [Common Criteria, сертификация дает право использовать продукт в госучреждениях США, – прим.ред.]. Даже по этим двум технологиям ясно, почему Fedora важна для Red Hat.

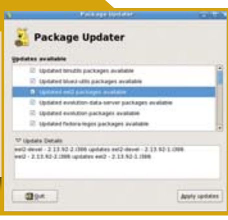
Так или иначе, Red Hat активно ищет способы прямого влияния на Fedora более широкого Linux-сообщества, особенно в областях, где Red Hat традиционно не сильна. Идеальным кандидатом для этого может быть KDE, и нам кажется, каждое подобное начинание Fedora может принести пользу.

Fedora 5 отнюдь не бедна исправлениями и новыми приложениями. В данном релизе уйма нововведений; из них интереснее всего, вероятно, Mono. Включили его почти в последний момент, в январе, в связи с озабоченностью расхождением путей Fedora и OpenSUSE по части инструментов разработки (*GCJ* и Mono соответственно). Теперь Mono интегрировано в структуру Fedora вместе с Beagle, занявшим почетное место в меню Места (Places) в качестве функции поиска.

На просьбу прокомментировать причины обращения к Mono, представитель Fedora Грег ДеКёнигсберг (Greg DeKoenigsberg)

НА ПЕРВЫЙ ВЗГЛЯД

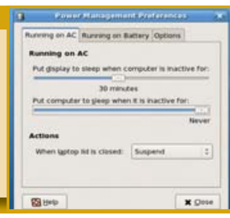
Этот шустрый инструментик найдет вам все нужные обновления.



Полная боеготовность: *NetworkManager* поможет отыскать все точки web-доступа.



Что делать, если погас свет? Спросите Gnome и его мудрый *Gnome-power-manager*.



просто озвучил официальную позицию, гласящую: «Деловые соображения, не позволявшие включать в Fedora некоторые компоненты Mono, исчерпаны». Что бы за этим ни стояло, для сообщества Fedora новость все равно хорошая. Также включены Mono-приложения *Tomboy* и *F-Spot*. Даже непонятно, как мы раньше управлялись без этой записной книжки и первоклассного менеджера фотографий.

GL на рабочем столе

AIGLX – еще одно важное дополнение в FC5: попытка Fedora обеспечить приятный вид и аппаратное ускорение рабочему столу *Gnome/Metacity*. Однако оно поддерживает скудный набор видеокарт, и на данный момент его применение ограничено. Как ни странно, дружелюбная к Linux Nvidia пока не поддерживает *AIGLX*. Можно удивиться, почему Fedora включает технологию, конкурирующую с *Xgl* от OpenSUSE, но на самом деле разработчики используют преимущества модульности *X.org 7.0* и предлагают функции, которые, возможно, будут включены в следующий релиз *X.org*, а не выкатывают очередной *X* сервер.

Приятно было увидеть и две новых утилиты мониторинга *Frysk* и *System Tap*. Хотя они все еще в ранней стадии, развиваются обе очень бурно. Frysk позволяет пользователю анализировать отдельные процессы и потоки, предоставляя прозрачный доступ к происходящему на низком уровне, а *System Tap* помогает отлавливать узкие места в производительности, тормозящие систему. Обе утилиты обозначены в данном релизе просто как презентация технологии, но в FC6 и FC7 следует ожидать от них большего.

Апплеты повсюду

Конфигурирование и управление пакетами – еще одно свидетельство, что сотрудники последние девять месяцев не мыльные пузыри пускали: здесь есть чем восхититься. Начнем с того, что беспроводные подключения и их использование теперь еще проще, благодаря паре добавленных в дистрибутив новых апплетов и включению драйверов ipw2200 и bcm43xx в ядро. Апплет *NetworkManager* занимается сетевым доступом: помогает управлять имеющимися подключениями и автоматически определять локальные беспроводные сети. Поддерживая шифрование WEP и WPA/WPA2, а также модули для VPN-соединений Cisco, *NetworkManager* очень украшает Fedora.

Другой апплет, для *Gnome-power-manager*, является большим шагом в верном направлении для управления питанием в Fedora. Включенный как презентация технологии, он позволяет настраивать реакцию системы на различные события, связанные с питанием. И неудивительно, ACPI все еще своенравен, и режим suspend в нашем ноутбуке Dell работает некорректно.

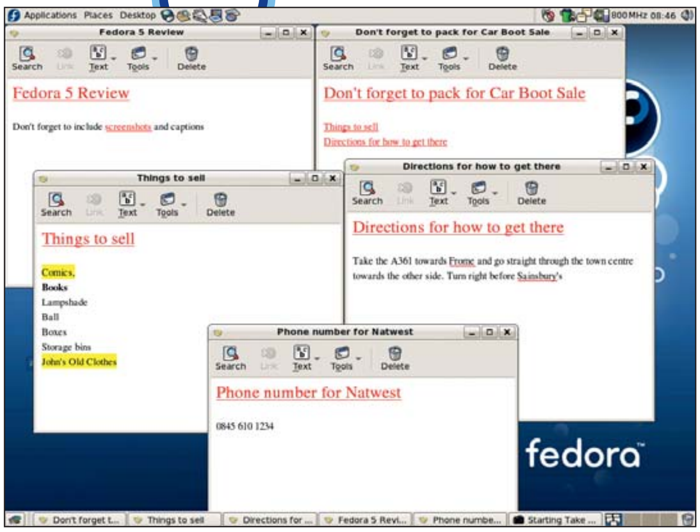
Теперь вкусности *Yum*. Пользователям Fedora, конечно, известно, что *Yum* служит менеджером пакетов еще с FC1. Теперь он объединен еще с двумя утилитами: *Pup* и *Pirut*. Изумительно названный *Pup* (щенок) обновляет пакеты. Он чрезвычайно прост в использовании, но не имеет никаких настроек, а также не имеет системы уведомлений типа *Rhn-applet*, использующийся в RHEL, но это мелкий недостаток, а не просчет.

Pirut – прямая замена утилиты *System-config-packages*, и что это за замена! Согласитесь, нельзя было быть ограниченнее, чем *System-config-packages*: установка пакетов допускалась только с CD/DVD. Благодаря *Yum* вместе с *Pirut* и *Pup*, это ограничение снято. Сам *Yum* приобрел несколько новых возможностей; лучшая из них – поддержка подключаемых модулей. По умолчанию к *Yum* подключен модуль *installonlyn*, предотвращающий установку более двух ядер одновременно. Вы можете установить другие модули, наращивая функциональность *Yum*.

Более приятной новостью для нас является то, что утилита *System-config-language* позволяет теперь работать не только на английском языке. Читатели, возможно, помнят, что отсутствие альтернативы было небольшим, но важным минусом FC4, но сейчас он исправлен, на радость нашим коллегам из России. Отполированы другие инструменты настройки, и появился центр управления, в котором все они собраны под одной оболочкой. Центр управления еще в ранней стадии, но может оказаться полезным благодаря всеобъемлющему подходу.

Ай да релиз!

Со всеми этими дополнениями, Fedora теперь весьма увесистый дистрибутив (на пяти CD, в отличие от четырех предыдущей версии). Но наряду с блестящими новинками, видны разумные переделки, облегчающие жизнь пользователя. Например, кроме объединения с *Beagle*



Записки в Tomboy's можно связать друг с другом – теперь данные не потеряются!

в меню Места (Places), *Pirut* размещен в пункте **Добавить/Удалить приложения** (Add/Remove Software) меню **Приложения** (Applications), а не зарыт в меню **Системные Утилиты** (System Tools) или **Избранное** (Preferences). Подобные мелочи помогают новым пользователям Fedora быстрее разобраться с системой.

Попользовавшись дистрибутивом и пообщавшись с сообществом, мы убедились, что проекту Fedora суждена долгая жизнь. Мы заметили, что хотя в *Anaconda* (установщик Fedora) присутствуют важные внутренние изменения, они не полностью отражены в Fedora 5, и оставлены для доработки к FC6 или даже FC7. Проект на всю катушку использовал лишние три месяца, чтобы создать не просто отличный дистрибутив, но также и фундамент для будущих версий. Red Hat Enterprise Linux 5 появится в конце этого года, и огромные усилия по разработке Fedora 5, похоже, больше посодействуют RHEL 5, чем Fedora 6. Кстати, в Fedora 6 планируется вернуться к шестимесячному циклу, но разработчики от души насладились дополнительным временем, перепавшим на долю данного дистрибутива, и могут передумать.

Что ж, мы не торопимся. Совокупность нововведений и обновлений Fedora Core 5 с лихвой окупает дополнительное время разработки, и возвращает Fedora в стан соперников OpenSUSE и Ubuntu. LXF

ГРЭМ СЧИТАЕТ...



«Новый вид освежает, как и включение Mono, Beagle и *Xen*. А новый менеджер пакетов неуклюж и чересчур упрощен, что затрудняет поиск приложений.»

ОСНОВНЫЕ КОМПОНЕНТЫ FEDORA CORE 5

- Ядро 2.6.15
- Gnome 2.14
- KDE 3.5
- *Apache 2.2*
- *Beagle 0.2.1*
- *GCC 4.1*
- Mono 1.1
- *MySQL 5*
- *OpenOffice.org 2.0.2*
- PHP 5.1
- *Xen 3*

LINUX FORMAT TOP STUFF AWARD

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	10/10
УДОБСТВО ИСПОЛЬЗОВАНИЯ	9/10
ОПРАВДАННОСТЬ ЦЕНЫ	9/10

Лучший выпуск Fedora – значительные улучшения над FC4 убеждают, что он должен быть в вашем списке обновлений..

РЕЙТИНГ 9/10

СЕТЕВОЕ ОБОРУДОВАНИЕ

Devolo MicroLink dLAN Highspeed

Электросетевой Ethernet – равно как и **Ник Вейч** – получает столь необходимое увеличение скорости.

САМОЕ ГЛАВНОЕ

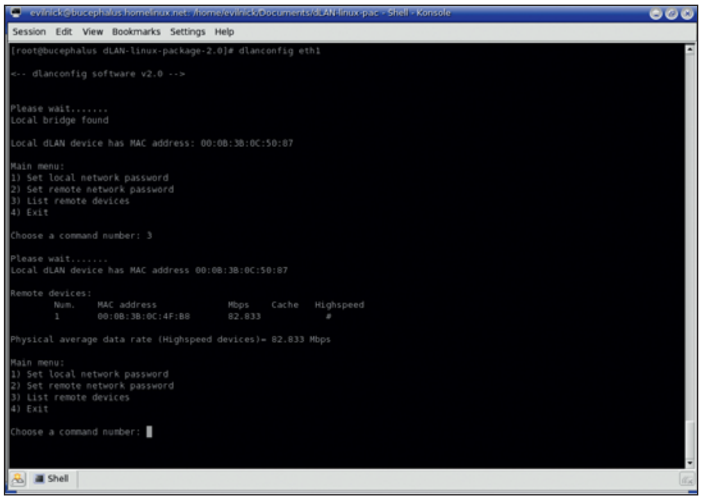
Эти устройства направляют сетевой трафик через электропроводку.
См. также: Wi-Fi или традиционные проводные сети..

- **РАЗРАБОТЧИК:** Devolo
- **САЙТ:** www.devolo.co.uk
- **ЦЕНА:** около $25 (за пару)

Соединение компьютерных устройств в вашем доме стало повседневной рутиной – сегодня нам подавай ноутбук, подключение игровой приставки... не говоря уже о моде на файл-сервера в некоторых шибко умных семьях. Devolo – одна из фирм, предоставляющих простую альтернативу сверлению новых дырок в стенах и опутыванию дома пучками кабелей: Ethernet через бытовую электросеть.

Мы рассматривали эту технологию в *LXF66*, когда описывали оригинальное dLAN-оборудование от Devolo. Вообще говоря, любое dLAN устройство напоминает блок питания. Оно вставляется в розетку, но вместо 12 вольт вы получаете Ethernet-соединение. Небольшой встроенный сетевой контроллер модулирует сетевые пакеты в сигналы, которые свободно могут быть переданы по существующим электропроводам и приняты таким же устройством на другом конце. Подобная технология передачи сигналов использовалась ранее для удаленного управления оборудованием, но новая техника открыла ее и для сферы высокоскоростных сетевых коммуникаций.

dLAN-адаптер версии 'Highspeed' выглядит также, как и обычный: небольшая полупрозрачная синяя коробка, размером с 12-вольтовый блок питания, который наверняка у вас уже есть. Устройство имеет одно гнездо RJ-45 для Ethernet-кабеля и несколько зеленых LED-индикаторов, спрятанных под пластиком; их мигание подскажет вам, идет ли передача, и есть ли соединение. Вот примерно и все о нем.



Прилагаемое ПО (включая драйвера и утилиты настройки) элементарно, но работоспособно – вам просто надо один раз запустить его.

Ускоряемся

Новый адаптер имеет слово Highspeed в названии, потому как рабочая ширина канала значительно увеличена. Старые устройства работали со скоростью 14 Мбит/с, что было вовсе не здорово, поскольку беспроводные устройства предлагали уже 54 Мбит/с. Зато новое устройство дает 85 Мбит/с, и, в отличие от Wi-Fi, эта скорость обычно достижима: при нашем тестировании на кабеле длиной около 50 м средняя скорость получилась 82 Мбит/с. Впечатляет. Конечно, это вся доступная полоса пропускания, то есть соединение будет поделено между всеми подключенными устройствами.

И все же, такой скачок скорости делает вполне реальными и передачу видео по сети, и уменьшение времени резервирования ваших файлов на центральном сервере. Устройство работает на расстоянии до 200 метров, хотя скорость может значительно снижаться за счет помех в основной сети или длины кабеля. Преимущества Wi-Fi никто не отменял, но Ethernet через электросеть дешевле, проще, а иногда и надежнее, особенно когда сетевое устройство является стационарным, как, например, игровая приставка или сервер.

Все, что потребуется сделать для настройки вашей сети, это воткнуть устройство dLAN в розетку, а сетевой кабель – в гнездо устройства. Проще некуда. Ну, если страдаете паранойей, можете еще установить пароль шифрования – для этого имеется готовое к загрузке программное обеспечение под Linux, заодно проверяющее настройки сети. Оно компилируется из исходных текстов, но больших проблем возникнуть не должно. После запуска консольное приложение можно использовать для опроса подсоединенных устройств и для установки пароля. Хотя соединение через электропроводку безопаснее, чем беспроводное, все-таки остается возможность, что при некоторых настройках ваши данные подслушает соседский утюг. Устройство dLAN HS предусматривает DES-шифрование всего трафика со специальным уникальным номером, предотвращающим утечку данных.

Адаптеры продаются парами или поштучно (одно устройство вам, естественно, ни к чему, но вдруг потребуется расширить уже работающую сеть). В больших сетях такое решение обойдется значительно дороже, чем сверление дыр в стенах для прокладки кабеля или установка Wi-Fi повторителей. Но dLAN-устройства можно использовать в составе смешанной сети: один поместить дома у маршрутизатора, а другой у сервера в гараже или у игровой приставки, а может, и в гостиной, если понадобится. Гибкая и полезная альтернатива Wi-Fi. LXF



Devolo говорит о теоретической возможности использования 253 адаптеров в одном доме. Но это может быть дороговато.

LINUX FORMAT TOP STUFF AWARD

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	10/10
УДОБСТВО ИСПОЛЬЗОВАНИЯ	10/10
ОПРАВДАННОСТЬ ЦЕНЫ	8/10

Ethernet через электропроводку – жизнеспособное решение, а набор Devolo – лучшее из того, что мы видели..

РЕЙТИНГ 9/10

ЦИФРОВОЙ ВИДЕОМАГНИТОФОН

MythTV 0.19

НА ДИСКЕ

Образовалась свободная минутка? Почему бы не найти занятие получше, чем смотреть телевизор, спрашивает **Грэм Моррисон**. Можно, например, настроить эту мультимедиа-штучку.

САМОЕ ГЛАВНОЕ

Превращает ваш ПК в многофункциональный медиа-центр и видеомагнитофон.
См. также *Freevo* или *GeexBox*.

- **РАЗРАБОТЧИК:** Айзек Ричардс (Isaac Richards) и команда разработчиков MythTV
- **САЙТ:** www.mythtv.org
- **ЦЕНА:** Бесплатно по лицензии GPL

Первой нашей реакцией на данную версию было: «Он стал тормозить». Это заметит каждый, кто знаком с предыдущей версией *MythTV* (0.18). *MythTV* – мощное приложение для записи видеопрограмм со встроенной базой данных, а так как при воспроизведении используются буферизованные потоки данных, то управление обязано работать быстро. Но за шесть месяцев, прошедших после выпуска предыдущей версии *MythTV*, разработчики столько всего добавили, что на переключение каналов вместо умеренных двух секунд понадобится почти шесть. Поэтому менять телевизор на *MythTV* пока не хочется.

Правда, дополнительные задержки – результат нововведений. *MythTV* теперь может записывать любую просматриваемую вами программу и держать ее в буфере до семи дней. Если это лучшая серия любимого сериала и вы хотите хранить ее подольше, то, досмотрев ее, нажмите клавишу R. Можно также просматривать программы из буфера путем изменения фильтра в окне Просмотр записей (Watch Recordings).

Если не менять настройки автоудаления, то при переполнении диска *MythTV* удалит из буфера все программы без разбора (включая сохраненные). Получается, что диск всегда полон; у тех из нас, кто привык управлять жестким диском вручную, это не вызовет восторга. Такой подход использует жесткий диск по максимуму, но лучше бы в первую очередь удалялись уже просмотренные программы.

Новинка данной версии – мониторинг сигналов карт DVB и pcHDTV (телевидение высокой четкости), но он сыроват: правильные настройки нужно подбирать самостоятельно, а при некорректных установках найти что-нибудь мониторингом сигнала практически нет шансов. Данная функция намного лучше реализована в KDE'шном *Kaffeine*, там есть прекрасное окно со списком общих DVB-источников, а сканирование и добавление каналов происходит автоматически. *MythTV* далеко до простоты *Kaffeine*, но он на верном пути. DVB-записи теперь сохраняются в родном MPEG2 формате, и конвертировать и копировать на другие устройства стало намного легче.

Другой признак растущей зрелости *MythTV* – он больше не нуждается в сторонних источниках для получения информации о программе передач (в США обычно использовались XMLTV или www.tv.zap2it.com). Теперь можно загружать данные непосредственно из DVB-канала, а *MythTV* заполняет свою базу автоматически. Единственный минус – DVB-канал обычно дает информацию только на пару дней вперед (в отличие от 13 дней XMLTV).

Подключение и отключение

Есть несколько небольших, но важных изменений в подключаемых модулях, расширяющих функциональность *MythTV*. Воспроизведение DVD теперь осуществляется встроенными подпрограммами, а не *Xine* или *MPlayer*, а музыкальный проигрыватель способен передавать информацию на внешний ЖК-дисплей. Есть также совершенно новые модули, которые заинтересуют наших читателей в США: *MythFlix* позволяет использовать вашу регистрационную запись NetFlix для загрузки и онлайн-проката DVD.



Уж если *MythTV* заработал, то пользоваться им и правда просто.

Но главные улучшения видны в *MythWeb*. *MythWeb* – коллекция PHP-сценариев, позволяющих составлять расписание и просматривать записи в web-браузере, и это одна из лучших функций *MythTV*. Во многих случаях настраивать и управлять записями при помощи *MythWeb* удобнее, чем из основной графической оболочки, просто потому, что в окне браузера помещается больше опций, чем на экране телевизора. Основные изменения в переработанном интерфейсе пользователя коснулись индексной страницы, так что миниатюрные изображения записанных программ теперь загружаются намного быстрее, чем в предыдущей версии.

MythTV – превосходный продукт. Благодаря этому открытому, гибкому и нашпигованному функциями приложению есть смысл потрудиться и приставить ПК к телевизору. Если *MythTV* настроить правильно, он сможет производить автоматическую запись программ по расписанию и управлять вашей фильмотекой и музыкальной коллекцией, позволяя каждому в доме смотреть программы по своему выбору. Проблема всего одна: слова «если настроить правильно» в предыдущем предложении следует заменить на «если пожертвовать двумя месяцами жизни, чтоб заставить систему работать». Сказать, что *MythTV* трудно настроить, значит не сказать ничего: большинство из нас не спасет даже расширенная и улучшенная wiki. Приложению просто нельзя быть столь сложным, ведь это серьезное препятствие для большинства пользователей. И все равно это достойное обновление; если у вас есть время на настройку *MythTV*, то с ним мало что сравнится. LXF

ВЕРДИКТ LINUX FORMAT

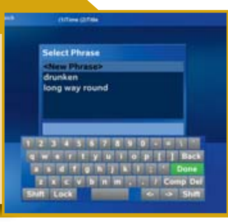
ВОЗМОЖНОСТИ	10/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
УДОБСТВО ИСПОЛЬЗОВАНИЯ	3/10
ДОКУМЕНТАЦИЯ	6/10

Запись по ключевым словам, пауза в живом эфире, многооконный интерфейс – *MythTV* все это делает, но ценой усложнения.

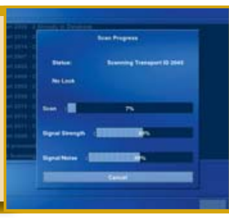
РЕЙТИНГ 7/10

НА ПЕРВЫЙ ВЗГЛЯД

Забудьте обычный пульт управления – при новом всплывающем окне с клавиатурой он вам ни к чему.



***MythTV* может автоматически сканировать цифровые ТВ-каналы, а также импортировать ваш собственный список.**



Новая тема MythCenter – лучшее во всем наборе.



ОКРУЖЕНИЕ РАБОЧЕГО СТОЛА

Gnome 2.14

Итак, Линус не любит Gnome. А что скажет **Энди Хадсон**?

САМОЕ ГЛАВНОЕ

Простое, но мощное рабочее окружение. См. также: KDE или Xfce.

- **РАЗРАБОТЧИК:** The Gnome Project
- **САЙТ:** www.gnome.org
- **ЦЕНА:** Бесплатно по лицензии GPL

Неужели Gnome 2.12 вышел всего шесть месяцев назад? Это был сногсшибательный релиз, и нас заинтересовало, чего проект достигнет в будущем, еще до того, как Линус Торвальдс (Linus Torvalds) разразился в своей рассылке провокационной тирадой о том, что использовать надо KDE. Его замечания, что разработчики Gnome – «нацисты от интерфейса», а пользователи Gnome – «идиоты», только разожгли интерес к тому, что представит команда в релизе 2.14.

Обычно обновление Gnome балует нас большим количеством изменений, и этот релиз не исключение. Самое важное – включение инструментария векторной графики Cairo (много позаимствовавшего из Qt'шного Arthur), придающего Gnome отполированный до блеска, по сравнению с предыдущими версиями, вид. Cairo – одна из причин, почему Linux (и в частности, Gnome) приобретает черты Mac OS X. Ступенчатая графика больше не режет глаз; плавность линий заставляет рабочий стол Gnome просто сиять.

Внешний вид и ощущения от Gnome практически прежние, тема по умолчанию Clearlooks держится с версии 2.12. Что до внутренних изменений, то, возможно, наиболее интересным является включение мощного инструмента поиска *Beagle* в пункт Поиск (Search) меню Places. Хотя *Beagle* не устанавливается вместе с Gnome по умолчанию, вы можете выбрать его в качестве настольной поисковой машины. Удивляться нечему: *Beagle* напрямую финансируется Novell, которая также делает ставку на разработку Gnome. Можно поспорить, хорошо ли это, но привлекательность Gnome от этого усилилась.

Предыдущий обзор Gnome восхвалял включение утилит администрирования, и 2.14 добавил к ним еще две: Sabayon и *Pessulus*. *Sabayon* – это необычная утилита конфигурирования оконного менеджера. Вместо обычного интерфейса с меню и опциями конфигурирования Gnome, он запускает вложенную *X*-сессию, и вы в реальном времени видите результаты изменений; администраторы будут довольны. *Pessulus* – редактор прав доступа, разрешающий или запрещающий функции Gnome-сессии. Обе утилиты предоставляют необходимую функциональность для использования Gnome в бизнесе.

Выходя в большой мир, вы насладитесь *NetworkManager*, который настолько упрощает настройку сети (особенно беспроводной), что кажется слишком хорошим, чтобы существовать в реальности. Мы использовали *NetworkManager* в hotspot'ах для подключения со смешанным доступом WEP и WPA, и он нас ни разу не подвел. Он также поддерживает VPN-доступ при помощи Vpnc, включенного в подменю VPN, и даже позволяет импортировать конфигурационные файлы pcf для Cisco VPN. Неплохо.

> «GNOME ПРЕДЛАГАЕТ КУДА БОЛЕЕ УДОБОВАРИМЫЙ СПОСОБ РАБОТЫ, ЧЕМ KDE.»

NetworkManager поощряет хорошую практику безопасности, помещая в черные списки стандартные SSID-имена от изготовителей, вроде belkin54g или linksys, и заставляя пользователей изменить их.

Достоинства

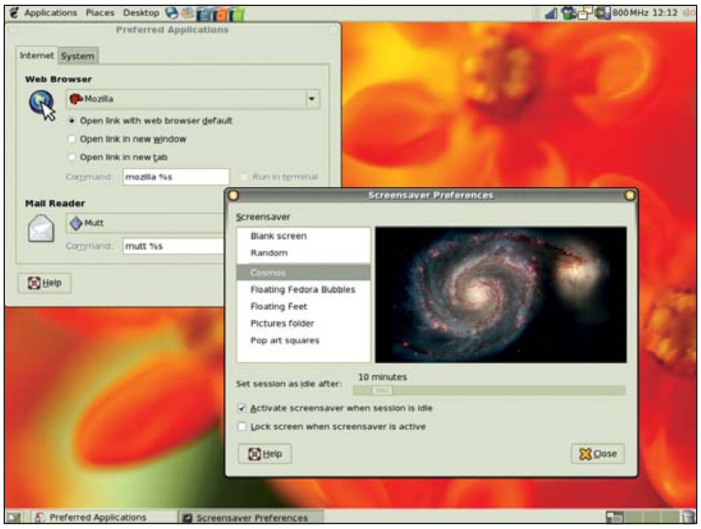
Когда Торвальдс критиковал упрощенность Gnome, ему пояснили, что разработчики делают рабочую среду, не пугающую новых пользователей. Такая философия подразумевает, что продвинутым пользователям, вроде Линуса, не всегда легко перестроить все под себя, но с другой стороны, пользователи, далекие от техники (например, жена автора обзора), могут запросто отыскать через Gnome дорогу к приложениям.

Недостатки

Единственный элемент, который мог бы быть дружелюбнее к пользователю, это приложение *Обо мне* (About Me), где хранятся личные данные. В частности, *About Me* – стандартный способ изменения пароля пользователя при помощи GUI, но необходимость запуска этого приложения для выполнения указанной операции вовсе не очевидна. Пользователи предпочтут открыть консоль и подать команду *passwd* – было бы лучше, если бы Gnome включил пункт Изменить пароль (Change Password).

Но это мелочь; в данном релизе есть реальные преимущества. Обновление не вызовет трудностей. Сам Gnome печально знаменит сложностью установки, так как его создатели представляют только tar-архивы, оставляя компиляцию пользователю. Но к моменту прочтения статьи вы уже найдете пакеты для вашего дистрибутива, а дистрибутивы последних выпусков, естественно, включают 2.14. Gnome достиг такого уровня зрелости, что хотя новые релизы и имеют новые функции, нет острой необходимости немедленного обновления, потому что и предыдущая версия достаточно хороша.

На горизонте уже маячит KDE 4.0, и команде разработчиков Gnome надо продолжать свою превосходную работу, чтобы следующая версия не уступала предложению от KDE. Тогда Gnome перешагнет многие другие оконные среды. Мы уверены, что пользователи Gnome понимают критику Линуса, но, увидев эту версию, вряд ли они с ним согласятся. LXF



Выбери хранитель экрана по вкусу – и наблюдай звезды. Gnome 2.14 очень похож на Mac OS X, отчасти благодаря Cairo.



ПОЛ СЧИТАЕТ…

«Sabayon – это же десерт! Дружелюбие к пользователю выражается легко запоминаемыми именами: Admin Tools гораздо понятнее, чем 'Pessulus'.»

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
УДОБСТВО ИСПОЛЬЗОВАНИЯ	7/10
ДОКУМЕНТАЦИЯ	9/10

Не ошибиться даже нарочно. Хотите приключений – установите его из исходных текстов.

РЕЙТИНГ 9/10

АНТИВИРУС

Sophos Anti-Virus 5

Грэм Моррисон тестирует новую Linux-версию одного из самых уважаемых антивирусов.

САМОЕ ГЛАВНОЕ

Централизованная защита от вирусов для сети из машин Linux. См. также: *ClamAV* или *F-Secure AV*.

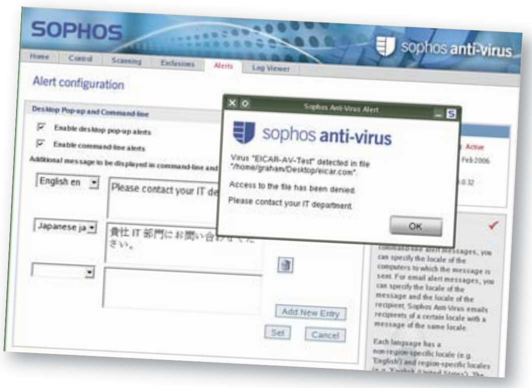
- **РАЗРАБОТЧИК:** Sophos
- **САЙТ:** www.sophos.com
- **ЦЕНА:** $300 для 5 пользователей с годичным обновлением

Sophos Anti-Virus был серьезным упущением нашего обзора в *LXF74/75*, просто потому, что тогда не было последней версии приложения, которую мы могли бы протестировать. Сейчас Sophos выпустила версию 5 своего маститого антивируса, включающую новую Linux-альтернативу.

Итак, Sophos присоединилась к когорте других поставщиков (Grisoft, F-Secure и Kaspersky) в попытке нажиться на распространенности Linux-серверов. Не является сюрпризом нацеленность и *Sophos AV* на корпоративный рынок. Это понятно и приятно: настольные приложения такого рода просто не нужны – по крайней мере, пока не наступит революция. На данном этапе достаточно проверять электронную почту и присылаемые файлы, а Linux-системы успешно с этим справляются. Linux – идеальная платформа для отлавливания заразы до того, как она причинит вред.



Вверху: **Отличное приложение, но web-интерфейс больше обещает, чем делает.**



Справа: **Если вам не повезло и вы подцепили вирус, сканирование по требованию сообщит об этом.**

Установка выполняется при помощи простого скрипта, и, получив ваши ответы на несколько вопросов, завершается автоматически. Прилагаются также бинарные модули ядра для большинства известных дистрибутивов Linux, но если ядро было обновлено, то в процессе установки потребуются исходные тексты ядра, а также *GCC 3.x*, чтобы скомпилировать модули для вашей системы. Увы, в журнале установки появляется обескураживающее заявление: *Sophos* оставляет за собой право не поддерживать приложение, собранное с использованием перекомпилированных модулей ядра.

Естественно, после установки антивирусного ПО первым делом обновляются вирусные базы. По умолчанию основное приложение обновляет их автоматически каждый час, но есть также небольшой инструмент командной строки для обновления вручную. Источник обновлений – на выбор: либо сервер Sophos, либо машина в вашей корпоративной сети. Последний вариант идет рука об руку с возможностью построить собственные пакеты при помощи скрипта установки, а затем легко распространить их по всей сети. После этого все машины пользуются одним репозитарием, поддерживающим базы в актуальном состоянии.

Удобные инструменты

После установки вы обнаружите стандартный набор консольных программ, которые научат ваши скрипты отслеживать вирусы, приобщившись к разносторонним талантам *Sophos*. Исключительно полный конфигуратор умеет менять все внутренние настройки. Можно, например, задать описание типов файлов, чтобы *Sophos* смог декодировать архивы и файлы других типов, плюс вести детальный протокол и указать общие пути.

Большинство пользователей, однако, увидят только web-интерфейс (доступный по адресу http://localhost:8081) – и очень жаль, потому что интерфейс не самый функциональный. Из него вы не сможете даже запустить простое сканирование на вирусы: это просто красивая панель для изменения малой толики из огромного числа важных настроек и просмотра отчетов. Будем надеяться, что в следующих версиях функциональности прибавится.

Наилучшая возможность *Sophos* – сканирование «на лету», по нашему мнению, одно из наиболее эффективных для настольного Linux. Любой файл, который вы попытаетесь открыть, локально или через Интернет, будет автоматически проверен. Вы и не заметите, как это произойдет; но если файлы большие и сканирование становится обременительным, можно указать в настройках типы файлов, которым вы доверяете. В противном случае *Sophos* обнаружит свое присутствие, только наткнувшись на вирус: при этом всплывает окно с сообщением и с советом обратиться к системному администратору. Приятно, что вы можете изменить текст всплывающего окна, указав список дальнейших действий.

Заразы не видно

Способы лечения зараженных файлов ограничены их удалением, или, если беда случилась с документом *Office*, удалением вредоносного прикрепления (это можно сделать из командной строки). На первый взгляд ограничение кажется серьезным: многие настольные антивирусы под Windows способны вытащить хоть какую-то информацию из зараженного файла. Но поскольку сейчас нет достойных упоминания Linux-вирусов, то подозрительные файлы в любом случае будут предназначены для других систем, и пока что этот подход оправдан.

В целом Linux-версия *Sophos AV* нам понравилась. Она использует ту же проверенную вирусную базу, что и другие продукты *Sophos*, а значит, всегда на передовой. Политика обновлений весьма разумная, а фоновый демон умел и эффективен. Хорошо бы web-интерфейс был более функциональным; но возможности вроде централизованного распределения пакетов делают *Sophos* хорошим выбором для малой локальной сети. LXF

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	5/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ДОКУМЕНТАЦИЯ	7/10

Мощная база данных и сканер по требованию, но маловато интеграции с Linux.

РЕЙТИНГ 7/10

СКАНЕР УЯЗВИМОСТЕЙ

Nessus 3.0

Как ни старался **Пол Хадсон**, ему не удалось заставить *Nessus* назвать его Дэйвом. Ясно, что искусственный интеллект – не главное нововведение, но, может быть, все-таки есть смысл обновиться до версии 3.0?

САМОЕ ГЛАВНОЕ

Популярный сканер уязвимостей, находит проблемы в сети и сообщает о них.
См. также: Sara, Nmap.

- **РАЗРАБОТЧИК**: Tenable
- **САЙТ**: www.nessus.org
- **ЦЕНА**: Бесплатно

Создатель, фирма Tenable, сообщила, что причиной закрытия стали конкуренты, которые перепаковывали исходный код *Nessus* и продавали его. Нам это кажется странноватым, поскольку мы всегда считали, что такая вещь в природе GPL, но в то же время известно, что *Nessus* не получал код от пользователей (или тех самых фирм), так что, возможно, мы сами во всем виноваты.

Сор в избе

Главное событие релиза – переписан движок сканирования, Tenable заявляет, что он быстрее, чем в версии 2.xx, и защищен лучше чем когда-либо. Система безопасности, хотя и не особо сложная, делает свое дело: дополнительные модули не могут открыть сокет или вообще подать голос, не будучи подписанными Tenable.

Насчет скорости, рады сообщить, что в этом релизе она значительно улучшена – сканирование нашей тестовой машины (Athlon 64 3400+, 1ГБ ОЗУ под управлением SUSE 10.0) *Nessus 3.0* с использованием 9,736 дополнительных модулей заняло всего 29 секунд, это куда быстрее, чем 1:01 минуты, потребовавшиеся для *Nessus 2.2* с 9,491 дополнительных модулей. С заданием просканировать все доступные порты *Nessus 3.0* справился за 43 секунды, а *Nessus 2.2* потратил на него 2:11 минуты.

Но скорость – это еще не все. При всех преимуществах *Nessus 3.0* над предыдущей версией, есть и плохие новости. Долгое время при составлении отчетов *Nessus* полагался на номера версий, и 3.0 – не исключение. Например, если, сканируя ваш ПК, *Nessus* обнаружит, что в системе запущен *Apache 2.0.48*, то перечислит в отчете все известные проблемы этой или более ранней версии. Ну а если в вашем дистрибутиве дыры залатаны без изменения версии? Вообще-то это стандартная практика. А *Nessus* плачется о проблеме, которой нет.

Заявлено, что в *Nessus* можно отключить мягкую проверку, тогда он будет искать реальные уязвимости вместо сканирования номеров версий, но, похоже, большой разницы не наблюдается. Когда мы попытались попробовать, то получили сообщение: «Возможна остановка или перезагрузка Windows» – есть чему удивиться, тестирование-то идет под Linux! Анализ номеров версий – лучший способ приумножить ложные срабатывания, и весьма печально, что *Nessus* не отказался от него.

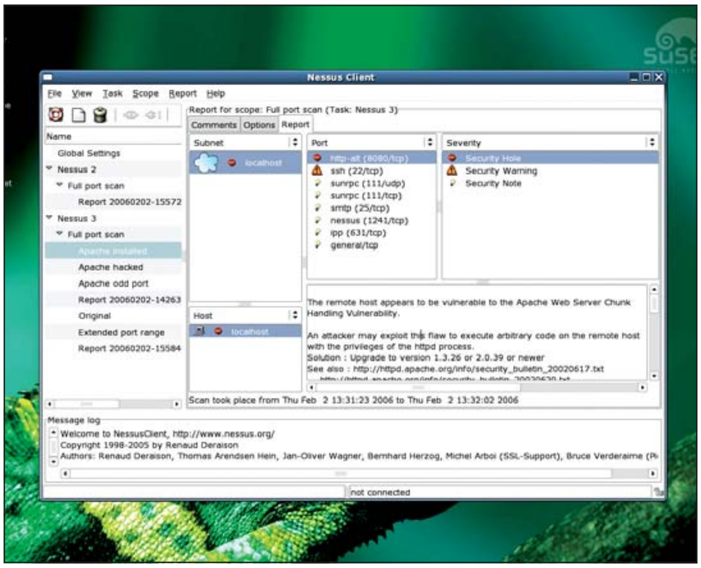
Огласите весь список!

Другая проблема – список портов, сканируемых по умолчанию: он включает лишь около 1/8 всех возможных портов. Скорость, что ли, оптимизировали? Выходит, если кто-то нелегально подключится к вашей машине через порт 50000, *Nessus* его не заметит. Правда, список сканируемых портов можно изменить, но это слабое утешение. Сканируя порты, *Nessus* не делает предположений о типе сервиса, так что при установке *Apache* на порт 5560 (если, конечно, указать этот порт для сканирования) *Nessus* догадается, что это *Apache*, и запустит соответствующую проверку. Зато если запустить демон *Nessus* на портах 6666 и 7777, Nessus 2.2 обнаружит их и сообщит, что они часто используются backdoor'ом NetBus, а *Nessus 3.0* просто известит, что порты открыты, без всякой дополнительной справки.

Обнаружив проблему, *Nessus* берется за дело. Прежде всего классифицируется тип проблемы (самое страшное – Дыра-Security Hole, затем идет Предупреждение-Security Warning, а наименьшая опасность – Замечание-Security Note). По возможности, *Nessus* дает исчерпывающее описание проблемы и возможное решение, и даже ссылки на предупреждения CVE (Common Vulnerabilities and Exposures – Общие уязвимости и подверженности, для непосвященных), так что вы можете изучить свои проблемы самостоятельно.

При всем при том, *Nessus 3.0* является важным обновлением программы, и похоже, что в будущем будет еще лучше. Tenable уже сулит учебные курсы и сертификацию по *Nessus* для администраторов, желающих упрочить свои знания и использовать больше продвинутых возможностей *Nessus*. Есть также изменения основной линии: Tenable обещает на будущее поддержку проверки на соответствие, которая позволит отсканировать один сервер и использовать его как образец для других, обеспечивая быстрое и простое отслеживание отклонений.

С открытым кодом или без, *Nessus 3.0* все еще лидер в своей области, и, кажется, останется им и далее. LXF



NessusClient, **бесплатный GUI, работает с версиями** *2.2* **и** *3.0* **– можно не учить все параметры командной строки.**



НИК СЧИТАЕТ...

«Не забудем простоту использования и толковые отчеты, которые сделали *Nessus* любимейшей утилитой безопасности в мире.»

ВЕРСИЯ 2.2 ЖИВА

Хотя 3.0 вышла с закрытым кодом, *Nessus 2.2* далек от гибели. Tenable даже выпустила несколько обновлений для пользователей 2.2, с кодом, доступным по лицензии GPL. Ясно, что версии 2.2 не видать нового быстрого движка или дополнительной безопасности ядра 3.0, но сцену она не покидает. Уже есть планы создать ответвление 2.2 с открытым кодом, и только время покажет, что из этого выйдет.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	8/10
ДОКУМЕНТАЦИЯ	6/10

Очень быстрый и мощный, но дважды проверьте, что тревога не ложная.

РЕЙТИНГ 8/10

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.

МИСТЕР УБУНТУ



Ладислав Боднар основатель, начальник и сотрудник сайта DistroWatch.com.

Недавно я встречался с Марком Шаттлвортом. Хотя наша беседа была краткой, я ещё долго находился под впечатлением от его визита. Помимо страстной речи, произнесённой им в местном университете, он рьяно участвовал в показательной установке, чтобы «увидеть, с какими проблемами сталкиваются люди во время установки Ubuntu».

Описывая эти события на DistroWatch.com, я обнаружил, что значительное число читателей выразило сомнения в искренности слов и действий Шаттлворта. Кое-кто зашел так далеко, что высмеивал популярность Ubuntu, якобы обусловленную глубокими карманами богатого учредителя, а не качеством продукта, который он столь щедро спонсирует. Даже исполнительный директор Mandriva намекнул, что Шаттлворт – человек с «почти бесконечной чековой книжкой», который в какой-то момент может захотеть и «выкинуть из бизнеса все остальные дистрибутивы, созданные сообществом, а потом начнет брать деньги с пользователей».

> «НЕТ ОСНОВАНИЙ ПОЛАГАТЬ, ЧТО У НЕГО ДУРНЫЕ НАМЕРЕНИЯ.»

Конечно, всем известно, что деньги и слава портят человека, однако нет оснований полагать, что основатель Ubuntu имеет дурные намерения. Он выполнил все свои обещания. Он разослал бесплатные компакт-диски тысячам пользователей в странах с плохой Интернет-инфраструктурой. Он даже создал десятимиллионный фонд поддержки Ubuntu, на случай, если угодит под автобус. Можно ли требовать большего? В конце концов, он мог бы «почти бесконечно» тратить свою чековую книжку на вино, блондинок и быстроходные катера...

Хороши весной дистрибутивы

Fedora Core 5 • SUSE 10.1 • Ubuntu 6.04 Упрощённая установка и трёхмерные рабочие столы

В связи с выходом трёх больших релизов от Fedora, SUSE и Ubuntu, первая пара месяцев 2006-го года характеризовалась интенсивной разработкой. Обещания появления новых крутых эффектов, повышения удобства и приведения в порядок рабочего стола Linux раздразнили интерес СМИ, посвященных технологиям. Может быть, рабочий стол Linux впервые сможет посостязаться со своими солидными конкурентами не только по цене и философии, но и по возможностям и удобству?

В соответствии со сроком выпуска, заявленным на 20 апреля [пока верстался номер, разработчики Ubuntu перенесли дату релиза на 1 июня, – прим.ред.], разработчики Ubuntu заморозили процесс добавления новых возможностей в версию 6.04 (Dapper Drake). В нее войдут графический инсталлятор *Espresso*, используемый в Live CD Ubuntu. Благодаря *Espresso* пользователям больше не нужно скачивать устанавливаемый компакт-диск, они могут взять Live CD, загрузиться с него и просто скопировать его содержимое на жёсткий диск.



Рабочий стол в виде трёхмерного куба – только одна из многих возможностей *Xgl* и *Compiz* от SUSE Linux 10.1.

В Dapper Drake также будет улучшен загрузчик (теперь он позволит пользователям выполнить тест памяти, проверить компакт-диск на наличие дефектов или восстановить повреждённую систему Ubuntu), подчищен экран входа в систему *GDM*, добавятся новые обои и иллюстрации и, конечно, новый с иголочки Gnome 2.14.

Большие новости в Novell – выпуск улучшенных графических подсистем Xgl и *Compiz* и оконного менеджера. Это событие весьма взволновало сообщество Linux, несмотря на задержку релиза дистрибутива из-за переделки графической подсистемы и серьёзных изменений в системе управления пакетами. Однако было за что побороться. Пользователи скорее всего просто обалдеют от нового рабочего стола: на него не просто приятно смотреть, трёхмерное представление поможет лучше визуализировать и организовать рабочее пространство пользователя. Его единственный недостаток – высокие требования к процессору и графической карте.

В отличие от Novell, разработчики Fedora решили не адаптировать *Xgl* и *Compiz*, а развивать свою собственную трёхмерную технологию *AIGLX*. Пятая версия Fedora не обещает особо серьёзных нововведений, но обновлённый *GCC 4.1* и *glibc 2.4* улучшат быстроту и безопасность дистрибутива.

• Читайте обзор Fedora Core 5 на стр. 10.

НОВОСТИ ОТ MANDRIVA

Из основных дистрибутивов только Mandriva пропускает гонку релизов во втором квартале. Это не значит, что разработчики дистрибутива валяют дурака: на самом деле бета-версия нового Live CD, названная Mandriva Linux One, была анонсирована в конце февраля. В отличие от Mandriva Move, известного проекта дистрибутива на Live CD, наиболее важная возможность в новом One – быстрая установка на жёсткий диск из Live CD.

Продукт основан на базе Mandriva Linux 2006 и содержит все исправления ошибок и улучшения безопасности, вышедшие с момента его выпуска, однако новых возможностей не добавлено и ни одно приложение не обновлено.

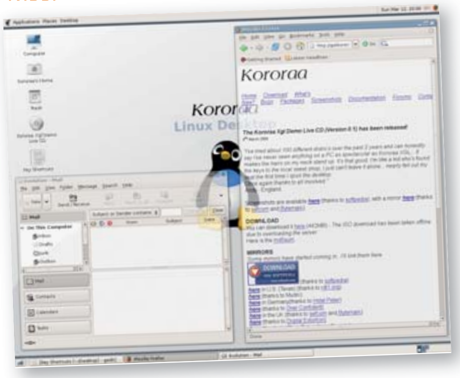
Возрождение Gentoo Linux

Gentoo Linux 2006.0 • Kororaa Xgl 0.1 Быстрые дистрибутивы

Релиз Gentoo Linux 2006.0 в конце февраля ознаменовался громким успехом. Хотя этот дистрибутив всегда был популярен среди продвинутых пользователей Linux, непростая процедура установки вкупе с усилиями, затрачиваемыми на поддержание пакетов в обновлённом состоянии, приводили к тому, что за сегмент начинающих пользователей и пользователей со средним уровнем опыта дистрибутив был вынужден бороться. Теперь же ситуация меняется. Благодаря последнему релизу Gentoo, привнёсшему полноценную функциональность Live CD, мощный графический инсталлятор и несколько интуитивных интерфейсов для управления пакетами, наблюдаются признаки всплеска интереса к этому дистрибутиву.

Прозрачность самого популярного дистрибутива на базе сборки пакетов из исходного кода привела к появлению производных от него дистрибутивов. Один из самых свежих и наиболее интересных – Kororaa. Созданный Крисом Смартом (Chris Smart) и Мэтью Оливером (Matthew Oliver), этот небольшой австралийский проект сделал себе громкую славу после того, как интегрировал *Xgl* и *Compiz* от Novell в первый в мире Live CD с Xgl.

Названный **Kororaa Xgl**, этот продукт позволит любому пользователю с относительно быстрым процессором (Intel Pentium 3 и выше) и 3D-акселератором от Nvidia или ATI насладиться невероятными визуальными эффектами, реализованными благодаря новому ПО. Хотя первый релиз Kororaa Xgl предоставляет рабочий стол только на базе Gnome 2.1.2, версия с рабочим столом на базе KDE уже разрабатывается и, возможно, будет доступна ко времени прочтения этих строк. Дополнительную информацию можно получить на домашней странице проекта. Кстати, «kororaa» на языке маори означает «пингвинчик».



Полупрозрачные окна – только один из многих эффектов нового дистрибутива Kororaa Xgl Live CD на базе Gentoo..

http://kororaa.org
www.gentoo.org

ХИТ-ПАРАД ДИСТРИБУТИВОВ

10 самых посещаемых страниц на DistroWatch.com в феврале
(среднее число визитов в день)

1	**Ubuntu Linux**	2,618	↔
2	**SUSE Linux**	1,901	↔
3	**Fedora Core**	1,494	↔
4	**Mandriva Linux**	1,169	↔
5	**Gentoo Linux**	869	↑
6	**Mepis Linux**	861	↓
7	**Damn Small Linux**	836	↓
8	**Debian GNU/Linux**	785	↓
9	**Knoppix**	717	↑
10	**Slackware**	612	↑

DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве визитов на страницы сайта, посвящённые конкретным дистрибутивам. Хотя эти цифры и не отражают настоящее количество инсталляций, они являются индикатором популярности дистрибутива на данный момент времени.

ЧЬЕ СООБЩЕСТВО АКТИВНЕЕ ВСЕХ?

Дистрибутив Linux (и не только Linux) – не просто строки кода. Это живая субкультура, направляемая тысячами разработчиков и пользователей, которые общаются через списки рассылок и форумы. Но много ли здесь реального взаимодействия? Мы подсчитали число постов в основных англоязычных списках рассылки за январь 2006-го года, чтобы понять, насколько активны пользователи того или иного дистрибутива. Хотя многие дистрибутивы предоставляют большой набор списков рассылок и форумов для различных языков, число постов является интересным индикатором уровня общения между разработчиками и пользователями.

Slackware Linux не включён в таблицу, поскольку не предоставляет списка рассылки, ориентированного на пользователя.

Дистрибутив	Имя списка	Кол-во постов (январь 2006)
Fedora Core	fedora-list	5,076
SUSE Linux	suse-linux-e	4,451
Debian GNU/Linux	debian-user	3,675
Ubuntu Linux	ubuntu-user	3,642
Gentoo Linux	gentoo-user	2,605
Mandriva Linux	newbie + expert	1,932
FreeBSD	freebsd-stable	885

Дистрибутив за $6.4 миллиона

RPath Linux 1.0 финансируемый венчурным фондом, предоставляет новые возможности для разработчиков

В ситуации, когда новые дистрибутивы прут как грибы после дождя, вполне простительно не заметить RPath Linux – ещё одну попытку вскочить на поезд Linux. Однако есть ряд причин, благодаря которым не заметить его нельзя. Во-первых, дистрибутив разрабатывается очень опытными программистами, которыми руководит Эрик Троан (Erik Troan), бывший вице-президент по техническим вопросам Red Hat, а также Билли Маршал (Billy Marshall) – бывший вице-президент Red Hat по продажам. Вдобавок RPath недавно получил финансовую поддержку от венчурного фонда в размере 6.4 миллиона долларов, на развитие рынка своих продуктов.

Самые интересные технические возможности RPath Linux – *RBuilder* и *Conary*. *RBuilder* – это инструмент совместной разработки в духе SourceForge: полная инфраструктура по разработке и поддержке специализированных дистрибутивов. *Conary* – технология управления ПО для дистрибутива, отдалённо напоминающая систему контроля версий (CVS) и сулящая упрощение поддержки индивидуально настроенных Linux-систем.

Первая стабильная версия RPath Linux была выпущена в феврале, после почти 18-месячной интенсивной работы. Хотя дистрибутив может существовать как отдельно установленный продукт, отсутствие какого-либо упоминания бренда предполагает, что он задуман скорее как базовая система для разработчиков, желающих использовать технологии Conary и *RBuilder* для создания собственных дистрибутивов. Один из таких проектов, по имени **Foresight Linux**, оказался в ряду наиболее приятных сюрпризов прошлого года, в основном за отшлифованный и самый свежий рабочий стол Gnome и приложения типа *Beagle*. LXF



Соучредитель и главный технолог Эрик Троан (Erik Troan) покинул Red Hat ради создания RPath.

www.rpath.com
www.foresightlinux.com

VIDALINUX – GENTOO ДЛЯ НАЧИНАЮЩИХ

Анатолий Евдокимов с удовольствием представляет вашему вниманию еще одну попытку сделать Gentoo Linux ближе рядовому пользователю.

Эту статью я решил посвятить замечательному пуэрториканскому дистрибутиву VidaLinux 1.2 (1.2.1) AMD64 (далее VLOS). В ней я постараюсь объяснить, что такое VLOS, чем он хорош для новичков и как произвести его первичную настройку.

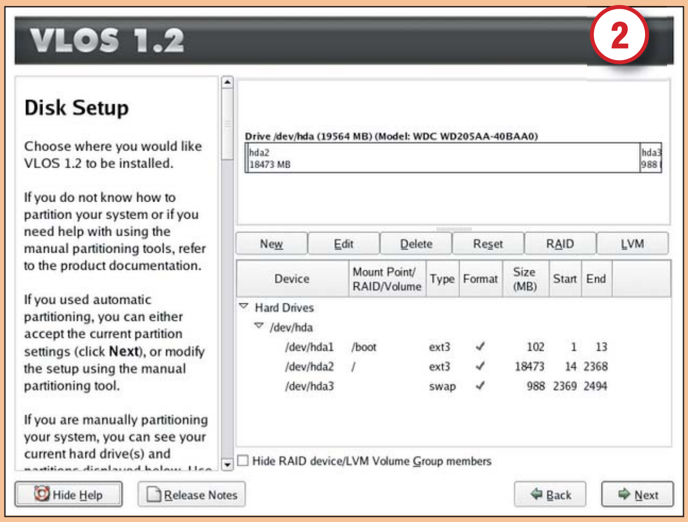
Несмотря на наличие огромного количества статей, FAQ и прочей справочной информации, человек, который решает установить Linux на своей машине, сталкивается со множеством проблем, ведь в сравнении с Windows это – другой мир, своя философия, которую трудно вот так понять с наскоку. И первая проблема – это, конечно, выбор дистрибутива. На данный момент все дистрибутивы можно разделить на две группы: бинарные дистрибутивы, в основе которых лежат уже скомпилированные пакеты и дистрибутивы на базе исходного кода (source-based), которые позволяют создавать более гибкие и оптимизированные системы, но, как правило, требуют больше времени и знаний. Пожалуй, самым известным представителем source-based дистрибутивов является Gentoo. Начинающего пользователя данный дистрибутив может отпугнуть сложностью установки и настройки; даже появление графического инсталлятора в версии 2006.0 не вполне решает эту проблему. VLOS – дистрибутив, основанный на Gentoo, но, в отличие от своего родителя, обладающий графическим инсталлятором Anaconda, который без особых трудностей позволяет установить базовую систему, и набором самых необходимых программ, что обеспечивает возможность приступить к работе без подключения к Интернету сразу после непродолжительной установки.



УСТАНОВКА VIDALINUX – ПОШАГОВАЯ ИНСТРУКЦИЯ



Шаг 1. Выбираем раскладку клавиатуры.



Шаг 2. Разбиваем жесткий диск с помощью *DiskDruid*.

ПРОВЕРЬТЕ ВАШ ДИСТРИБУТИВ!

emerge --sync – обновление дерева программ до текущего состояния;
emerge -av program – выводит список необходимых приложений и библиотек, которые нужно установить для использования program. В конце вывода добавляется вопрос: «Хотите ли вы установить данное приложение?». Если вы нажмете клавишу "y", *emerge* скачает и установит необходимые файлы;
emerge --resume – позволяет продолжить последний процесс установки приложения после сбоя;
emerge -b program – после успешной установки приложения система создаст бинарный пакет, чтобы не пришлось повторно компилировать файлы;
emerge unmerge program – удаление пакета из системы;
emerge -uD world – обновление всей системы.

Как видно из рисунков, установка VLOS достаточно проста и вряд ли может вызвать какие-то вопросы. Единственное, о чем стоит упомянуть, так это то, что программа разбивки жесткого диска *DiskDruid* позволяет отформатировать разделы лишь в ext2 и ext3. Не хочу сказать, что это плохие файловые системы, однако если вы хотите работать с более быстрой и современной ФС, преобразуйте раздел в ReiserFS. Для этого необходимо загрузиться в режиме спасения (F5) с установочного диска и в командной строке дать команду **mkreiserfs /dev/hdaX**, где X – номер раздела, который вы отвели для установки Linux.

Установка и удаление программ

Итак, система установлена – настало время доводить ее до ума. Прежде всего необходимо разобраться, как устанавливаются программы в вашей новой системе. Применительно к VLOS это значит хотя бы базовое знакомство с системой *Portage*.

Portage – это система распространения, сборки и установки программного обеспечения Gentoo Linux (а значит, и VLOS). Для того, чтобы получить последние версии программ во VLOS, достаточно ввести одну команду: **emerge --sync**. Она заставляет *Portage* обновить ваше локальное «дерево портежей» через Интернет. Ваше локальное «дерево портежей» содержит полную коллекцию сценариев, которые используются для сборки и установки последних версий пакетов Gentoo.

Но система *Portage* была бы ничем, если бы не замечательная утилита *emerge*, являющейся, по сути, главной программой в нашем дистрибутиве. В простейшем варианте ее вызов выглядит так: **emerge имя программы**. Утилита проверяет все зависимости и, в случае необходимости, скачивает недостающие библиотеки из Интернета, после чего компилирует и устанавливает выбранную программу. *emerge* обладает огромными возможностями, которые доступны через различные ключи командной строки. Полный список можно получить, вызвав **emerge --help**. Самые распространенные варианты можно найти во врезке.

Кроме того, перед вызовом emerge можно добавить флаги компиляции, например:

```
USE="mmx 3dnow" emerge xmms.
```

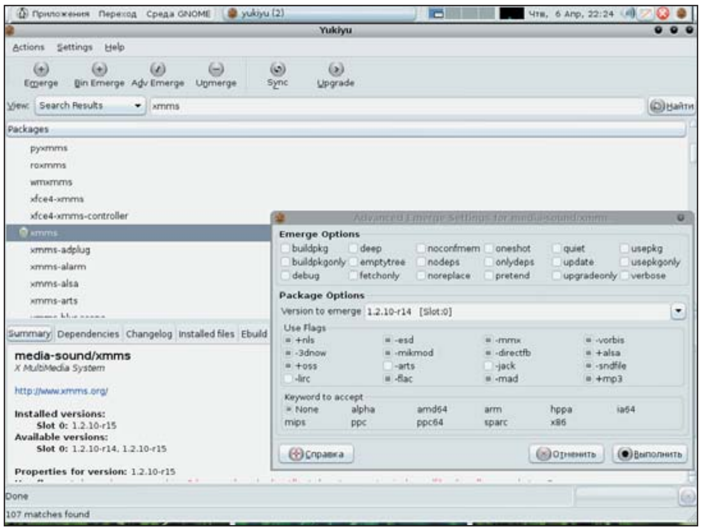
Поскольку приложения во VLOS собираются из исходных текстов, то, как и при ручной сборке, с помощью этих флагов мы можем включить или отключить (**USE="-3dnow" emerge xmms**) те или иные опции в собираемой программе, а с помощью строки **ACCEPT_KEYWORDS="amd64"** (которая также пишется перед emerge) оптимизировать программу под конкретную архитектуру процессора, что позволяет более тонко настроить систему в целом.

Данные базовые знания, по-моему, необходимы тем, кто будет работать с этой системой, поэтому я посчитал необходимым поделиться ими, несмотря на то, что во VLOS существует графическая оболочка к утилите *emerge* – YUKIYU. С ее помощью вы без особых проблем сможете установить, все что хотите, а также определить настройки глобальных переменных для установки пакетов.

Научи его говорить по-нашему

Разобравшись с принципами установки программ (а я надеюсь, мы это сделали), приступим к русификации системы. К сожалению, вывод команды **locale -a | grep ru** далек от идеального (кстати, в версии i686 ситуация немного лучше). Насколько мне известно, glibc давно уже собирается с поддержкой ru_RU.UTF-8. Но, увы, команда, позволяющая определить наличие в системе необходимой нам локали, свидетельствует о том, что это не так.

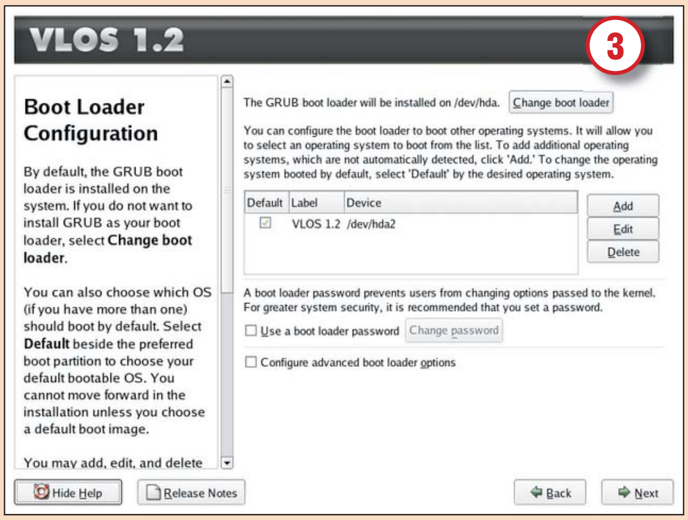
Ну что ж, это не проблема, сейчас мы достаточно быстро все исправим. Первым делом необходимо установить необходимые нам шрифты:



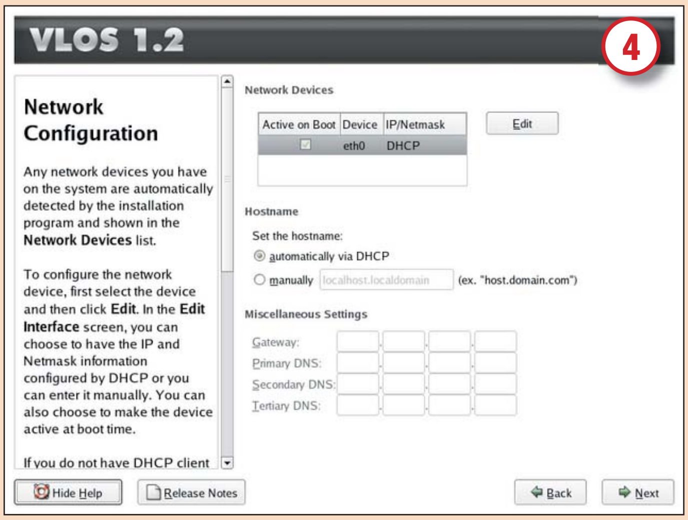
***Emerge* может выглядеть очень привлекательно.**

>>

УСТАНОВКА VIDALINUX (ПРОДОЛЖЕНИЕ)



Шаг 3. Устанавливаем загрузчик.



Шаг 4. Конфигурируем сеть.

<<

```
emerge terminus-font
```

Я бы еще посоветовал установить пакет mscorefonts – см. стр. 72.

Далее, создайте Unicode-локаль командой **localedef**. Для этого вам потребуются права root.

```
localedef --no-archive -c -i ru_RU -f UTF-8 ru_RU.utf8
```

Обратите внимание на опцию **--no-archive** она указывает на то, что локаль не надо упаковывать в архив. Если вы забудете указать эту опцию, то после обновления библиотеки *glibc* вам придется повторить этот шаг снова.

В каталоге **/etc/env.d/** создайте файл **02locale** (**touch 02locale**) следующего содержания:

```
LC_ALL=“”
LANG=ru_RU.UTF-8
```

Далее выполните следующие манипуляции:

В файле: **/etc/rc.conf**

```
UNICODE=“yes”
```

В файле: **/etc/conf.d/consolefont**

```
CONSOLEFONT=“ter-k14n”
```

В файле: **/etc/conf.d/keymaps**

```
KEYMAP=“-u ru4”
SET_WINDOWKEYS=“yes”
DUMPKEYS_CHARSET=“koi8-r”
```

В файле: **/etc/locales.build**

```
ru_RU.UTF-8/UTF-8
ru_RU.KOI8-R/KOI8-R
en_US/ISO-8859-1
en_US.UTF-8/UTF-8
```

Заключительным аккордом дожен стать перезапуск сервисов *keymaps* и *consolefont*.

```
/etc/init.d/keymaps restart
/etc/init.d/consolefont restart
```

Теперь у вас должна быть локаль ru_RU.UTF-8:

```
locale
LANG=ru_RU.UTF-8
LC_CTYPE=“ru_RU.UTF-8”
LC_NUMERIC=“ru_RU.UTF-8”
LC_TIME=“ru_RU.UTF-8”
LC_COLLATE=“ru_RU.UTF-8”
LC_MONETARY=“ru_RU.UTF-8”
LC_MESSAGES=“ru_RU.UTF-8”
LC_PAPER=“ru_RU.UTF-8”
LC_NAME=“ru_RU.UTF-8”
LC_ADDRESS=“ru_RU.UTF-8”
LC_TELEPHONE=“ru_RU.UTF-8”
LC_MEASUREMENT=“ru_RU.UTF-8”
LC_IDENTIFICATION=“ru_RU.UTF-8”
LC_ALL=
```

Остается еще настройка **gtk1+**. Дело в том, что после перехода на *X.Org* версии 6.8.2 абсолютно у всех пользователей с кодировкой ru_RU.UTF-8 появились проблемы с отображением русских букв: либо квадратики, либо закорючки, потому что нашу локаль просто забыли положить в **/usr/lib/X11/locale/**. О корректном решении данной проблемы можно прочитать здесь **http://wiki.fantoo.ru/index.php/HOWTO_GTK1_with_UTF8**

Но и это еще не все! Необходимо пересобрать пакеты *ncurses* и *slang* с опцией USE=“unicode”, а также в файле **/etc/make.conf** добавить LINGUAS=“ru”. Только после этого систему можно считать абсолютно русифицированой и готовой работать и устанавливать программы с поддержкой русского языка (если таковая в них предусмотрена).

Run, Level, run!

Думаю, вам не нужно объяснять, что такое уровень исполнения (**runlevel**). Именно **runlevel** определяет режим работы системы Linux и определяет перечень служб, запускаемых автоматически при загрузке.

Управление службами VLOS (и Gentoo) осуществляется при помощью команды **rc-update** (от имени root):

```
rc-update -s – показывает список всех запущенных служб.
rc-update -a – добавляет службу
rc-update -d – удаляет службу
```

Как видите, все просто. Попробуем применить эти знания на практике. В качестве звуковой подсистемы в *Vidalinux 1.2* используется OSS. Как перейти на прогрессивную ALSA? Правильный ответ: установить необходимый пакет (**emerge alsa**) дать команду **rc-update -a alsasound boot**. Количество автоматически запускаемых сервисов, как вы понимаете, определяет и скорость загрузки вашей системы. С помощью данной программы вы можете определить, что лично вам нужно, а что – нет, и тем самым выиграть пару секунд.

32-битное окружение

Поскольку в данной статье я по большей части говорю о версии для AMD64, то необходимо отметить что есть некоторые программы, которые работают исключительно в 32-битном окружении. Для того, чтобы обеспечить их выполнение, необходимо создать эмуляцию 32-битного режима в вашей 64-битной системе. Делается это достаточно просто: необходимо установить пакеты, отмеченные на рисунке справа.

Теперь ваша система сможет работать с самыми различными приложениями. Конечно, 100%-ной гарантии запуска дать нельзя, но исключения, все же, редки.

Контора пишет

Если для работы вам достаточно лишь читать разделы NTFS, то вам будет достаточно прочитать последний абзац данного раздела. В противном случае, т.е. при необходимости что-то записывать, ситуация осложняется, но выход, как всегда, есть.

В принципе, последние версии ядра Linux позволяют писать на NTFS-разделы, но в очень ограниченном объеме. Проект Captive NTFS (emerge captive) реализует другой подход – он использует родные драйвера Windows (в частности, **ntfs.sys** и **ntoskrnl.exe**) и эмулирует их окружение в Linux. В результате решается проблема закрытости спецификации NTFS – вместо инженерного анализа самой файловой системы мы используем для работы с ней родные драйвера, которые умеют делать это лучше всех (мне не удалось заставить работать Captive NTFS в 64-битной среде; есть информация, что пока это невозможно в принципе. К 32-битным системам это не относится).

После успешной установки необходимо предоставить требуемые файлы из состава Windows. Путей несколько:

1) Смонтировать обычным образом NTFS-раздел с установленной на нем Windows и запустить утилиту **captive-install-acquire**. Она произведет поиск по смонтированным разделам и скопирует все файлы, которые сможет найти.

2) Если какие-то файлы не могут быть найдены, но они точно есть – просто скопируйте их с NTFS-раздела в **/var/lib/captive**.

3) И последний случай: если нет установленной копии Windows, **captive-install-acquire** скачает с сайта Microsoft пакет XP SP1 и возьмет из него требуемые файлы.

После того, как все файлы будут на своих местах, можно приступать к монтированию. Незабудьте предварительно отключить «старые» NTFS-разделы.

```
mount -t captive-ntfs /dev/hda1 /mnt/win_c
```

Естественно, имя, раздел и точку монтирования нужно заменить на ваши. Все, теперь можно создавать, удалять файлы и вообще всячески безобразить на свежесмонтированных разделах.

Если же вас постигла неудача с установкой данного пакета, то, к сожалению, с возможностью записи и создания файлов на раздел NTFS формата лучше попрощаться. Вы по-прежнему сможете видеть этот раздел, читать и копировать с него различные файлы, чего, как вы понимаете, достаточно для просмотра фильмов, прослушивания музыки и прочих подобных операций. Внесите в файл **/etc/fstab** строку следующего вида:

```
/dev/hda5 /mnt/win_d ntfs ro,noexec,nls=utf8,auto,fmask=0111,dmask=0222 0 0,
```

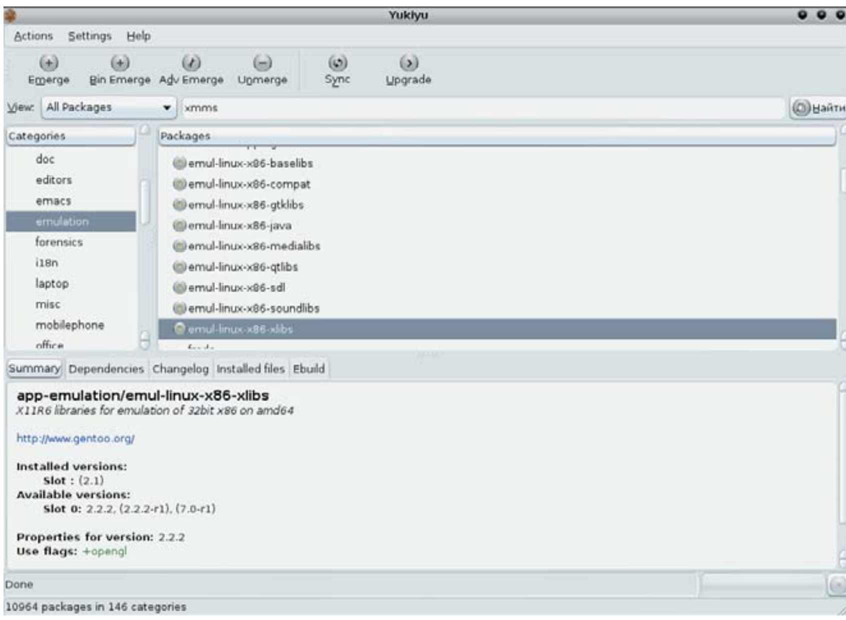
где **hda5** – ваш NTFS-раздел, **ntfs** – тип файловой системы, остальные опции определяют режим чтения, кодировку, способы отображения папок и файлов и т.д. Если вас интересует более подробная информация, то вы можете ее посмотреть с помощью команд **man mount** и **man fstab**.

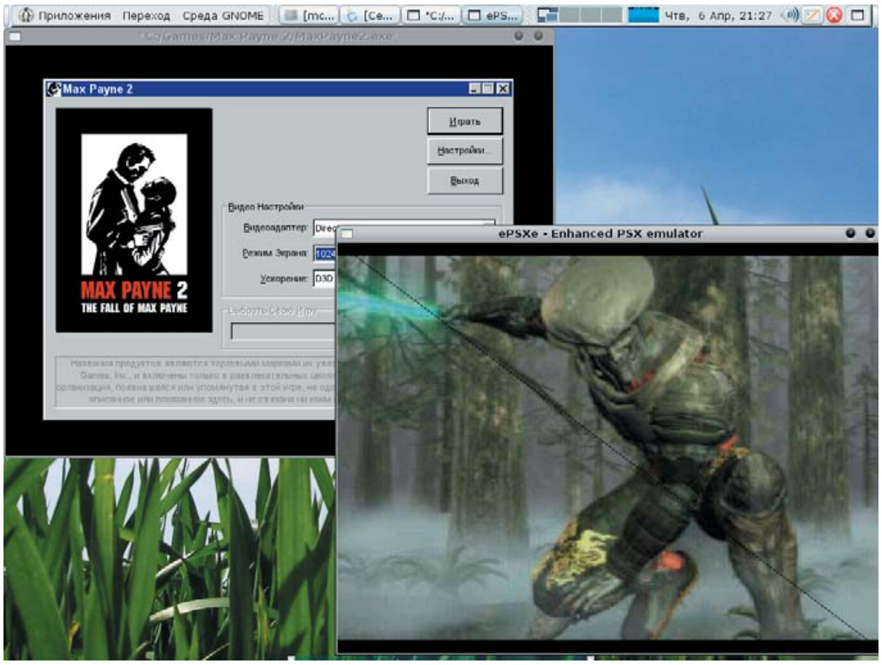
Немного об играх

Да, игры – это не последний момент для любой популярной пользовательской операционной системы, ведь каждый устает и ему хочется на досуге просто погонять какую-нибудь игрушку. Linux в этом отношении пока не в лидерах, и хотя ярые сторонники данной операционной системы могут сказать, что для нее написана куча игр [почти в каждом номере Linux Format мы рассматриваем одну-две лучших, – прим.ред], стоит сказать, что это в большинстве своем простые карточные или непривлекательные платформенные бродилки, а настоящих больших игр очень и очень мало. Однако нет безвыходных положений, есть отчаявшиеся люди. Ребята из компании TransGaming (**http://www.transgaming.com/**) явно не из последних. Благодаря им существует проект Cedega (ранее WineX), который позволяет запускать сотни Windows-игр.

Данный проект является коммерческим, поэтому для того чтобы получить доступ к программе, вам придется заплатить 5-10-

64 бита – это не всегда плюс.





Игры старые, игры новые.



VLOS становится полноценной игровой системой.

ДОМАШНЕЕ ЗАДАНИЕ

Пользуясь случаем, хочу порекомендовать некоторые приложения, которые, на мой взгляд, следует иметь любому пользователю:

gimp – аналог photoshop'а;

openoffice-bin – мощный офисный пакет (ставьте лучше бинарный пакет bin, иначе компиляция этого монстра может затянуться на сутки);

mplayer – мощный видеопроигрыватель (используйте лучше неофициальный ebuild http://bugs.gentoo.org/show_bug.cgi?id=92017 через overlay, это позволит вам получить интерфейс в Unicode. Неофициальный ebuild нужно скопировать в **/usr/local/portage/vlos/category/program/program.ebuild**. Затем создать дайджест файл **ebuild /usr/local/portage/vlos/category/program/program.ebuild digest** и далее устанавливать все как обычно);

xmms – музыкальный проигрыватель, аналог WinAmp;

songbird – тоже музыкальный проигрыватель, но с функционалом, близким iTunes (пока еще сыроват, поэтому потребляет очень много памяти, но в целом очень неплох);

firefox – популярный web-браузер;

f-spot (или qthumb) – просмотрщик графических файлов.

15 долларов, ну или собрать программу самим из исходных текстов, полученных через CVS. Во последнем случае рекомендую скачать скрипт http://cvscedega.linux-gamers.net/WineCVS.sh, запустить его (**sh WineCVS.sh**), в появившемся меню нажать g, затем 0 и, наконец, c. В открывшемся подменю вновь нажмите c еще раз для "checkout", затем нажмите r для запуска, затем 0. Похоже на шаманство, не правда ли? Не смущайтесь – до благополучного финала осталось совсем немного. Когда появится вопрос о директории, нажмите Y, чтобы создать её. Затем у вас спросят пароль – наберите «**cvs**». Нажмите c и r для компиляции, затем r для начала, затем выберите 0 для настройки. Наконец, скрипт начнёт конфигурирование и сборку.

Если же вы лицензионный подписчик – все намного проще. Вам достаточно скачать файлы **cedega-engine-5.1-local-update.i386.cpkg**, **cedega-5.1.tgz**, **cedega-small-5.1.tgz**. После чего скопировать последние два модуля в **/usr/Portage/distfiles** и дать привычную команду сборки: **emerge cedega**. Для того чтобы ваши игрушки бегали, а не ползали в духе любимой домашней черепашки, необходимо включить 3D-ускорение. Во VLOS это делается следующим образом: для NVIDIA – **emerge nvidia-glx**, далее в командной строке набираем **eselect opengl set nvidia** и заменяем **nv** на **nvidia** в файле **xorg.conf. ATI** – **emerge ati** и затем с помощью утилиты *fgrlx* производим необходимую настройку.

Итак, оснастку мы установили, но не забывайте про файл **cedega-engine-5.1-local-update.i386.cpkg** – это самое важное. Запустите Cedega, выберите в строке меню TransGaming (или alt+T), далее Install Local Update и укажите местоположение файла **cedega-engine-5.1-local-update.i386.cpkg** (последняя версия игрового движка от *Transgaming*, представляющая собой модифицированный особым образом wine). Все, теперь можно и поиграть!

Нажав Ctrl+C, мы можем вызвать панель управления профилями Configuration Profiles, которые используются играми. На базе профиля одной версии вы можете создать огромное количество модификаций, различающихся использованием той или иной версии Windows, настройками графики, звука и прочих полезных мелочей, обеспечивающих оптимальное качество игрового процесса.

Еще один вариант побаловать себя любимой игрой – это установить эмулятор какой-нибудь приставки. На мой взгляд, внимания и интереса достоин лишь ePSXe – эмулятор *Sony Playstation I*, остальное либо слишком старо, либо слишком сыро и нестабильно, как в случае с *psemu2* (эмулятор SPs 2). Для того чтобы установить ePSXe, вам потребуется размаскировать пакеты **epsxe160lin**, **gpupetemesagl176** и **spupeopsoss108**. Делается это достаточно просто – в файле **/etc/Portage/package.keywords** вы пишете категорию и пакет, который хотите размаскировать и в конце добавляете необходимое ключевое слово. Например **=app-emulators/epsxe160lin -***.

В результате ваш VLOS становится полноценной игровой системой.

Final Countdown

Ну вот, собственно, и все. LXF

LINUX FROM SCRATCH

С детства любили разбирать игрушки, чтобы узнать, что внутри?

Почему Linux From Scratch ?

Думаю, что если вы приступили к чтению статьи в названии которой присутствует фраза «Собери свой Linux», то вам не нужно рассказывать, что такое операционная система Linux, свободное ПО и так далее. Скорее всего, вы уже используете один из вариантов этой ОС на работе или на своем домашнем компьютере.

И, в зависимости от целей и разнообразных причин, как объективных, так и субъективных, вы, скорее всего, уже выбрали для себя один из так называемых дистрибутивов Linux. Дистрибутивом обычно называют подготовленный для работы в качестве единого целого набор из программы установки, системных утилит и служб, прикладных программ и модифицированного в соответствии с представлениями создателей дистрибутива, ядра Linux. С каждым дистрибутивом можно также связать определенную политику лицензирования и порядок технической поддержки. Из объективных причин выбора того или иного дистрибутива можно, например, назвать потребность в круглосуточной технической поддержке, а из субъективных – «навороты» дистрибутива, разрекламированные знакомым системным администратором, или внешний привлекательный вид рабочего стола.

По данным сайта *Distrowatch.com*, на сегодняшний день существует более 300 различных вариантов Linux, как общего назначения, типа Fedora Core, *Slackware* или Gentoo, так и ориентированных на решение специфических задач, например, *MoviX* или *Trustix*. И, по моему мнению, нельзя говорить о том, какой дистрибутив лучше, а какой хуже, безотносительно целей, квалификации пользователя, используемого оборудования и множества других факторов.

В этой статье я хотел бы поговорить о дистрибутиве *Linux From Scratch* (сокращенно *LFS*). Собственно, *LFS* – это даже не дистрибутив в классическом понимании этого слова. В *LFS* отсутствует одна важная деталь типичного дистрибутива Linux – программа установки.

Ее роль исполняет пользователь, вооруженный двухсотстраничной книгой – руководством к действию, которая как раз и называется *Linux From Scratch*. Собственно, помимо данной книги, написанной, кстати, Герардом Бикмансом (Gerard Beekmans) и доступной по адресу www.linuxfromscratch.org, «дистрибутив» состоит из архива исходных кодов плюс исправлений некоторых ошибок и, возможно, загрузочного LiveCD с заранее настроенной средой Linux, готовой для компиляции вашей собственной версии операционной системы из исходных текстов. Впрочем, доступный все на том же сайте www.linuxfromscratch.org загрузочный LiveCD вполне может быть заменен на уже установленный и соответствующим образом настроенный дистрибутив общего назначения. Требования и рекомендации к хост-системе (в которой пользователь собирает *Linux From Scratch*) также приведены в руководстве *Linux From Scratch* Бикманса (самое главное – ядро версии не ниже 2.6.2, скомпилированное GCC-3.0 или выше).

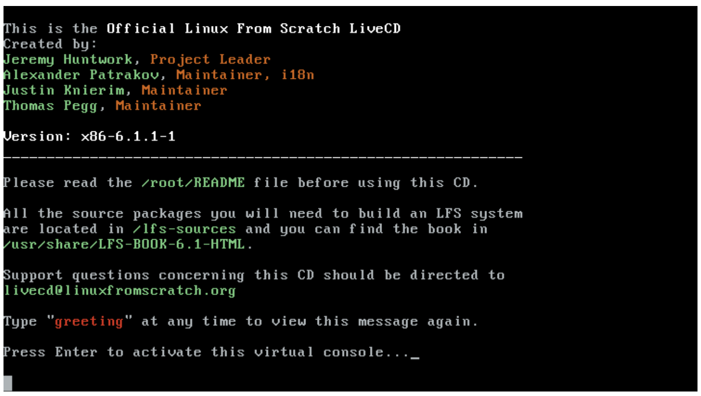


Рис. 1.

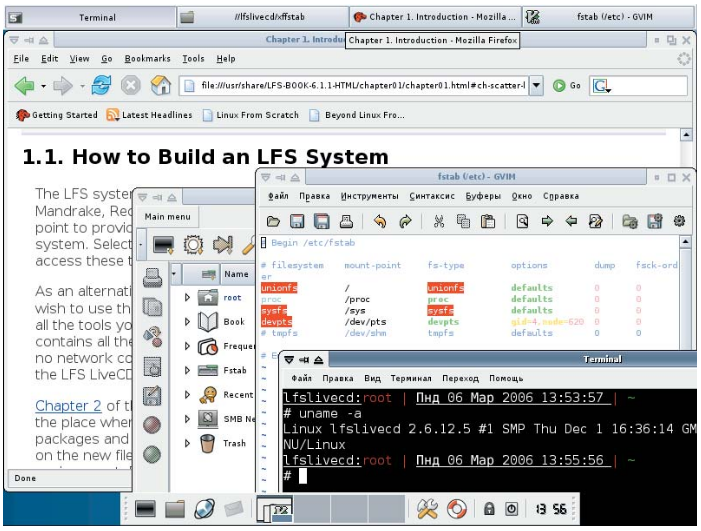


Рис. 2.

Хотя *LFS* вполне можно использовать и в качестве рабочего места разработчика или (после установки дополнительного программного обеспечения, не входящего в базовую систему *LFS*) сервера небольшого предприятия; возможно – даже рабочего места секретаря, все это не будет лучшим вариантом применения *Linux From Scratch*. Основная цель данного дистрибутива – познакомить пользователя с внутренним устройством операционной системы Linux. С помощью *LFS* мы можем по шагам пройти через весь процесс компиляции Linux, создания конфигурационных файлов, и наконец, спустя часы (или даже дни) работы, – загрузить операционную систему и увидеть на экране приглашение к вводу имени пользователя. Вдумчивое и внимательное выполнение инструкций, приведенных в руководстве *Linux From Scratch*, вместе с изучением документации по устанавливаемым в каждом шаге компонентам ОС позволит вам познакомиться с системой на уровне, не достижимом для пользователей и администраторов систем с закрытым исходным кодом. Сборка *LFS* может быть также весьма неплохой школой для технического специалиста или инженера, использующего коммерческие дистрибутивы Linux и поставившего себе цель понять процессы, происходящие в системе, на более глубоком уровне.

Еще одним применением *LFS* может стать задача оптимизации программного обеспечения операционной системы под конкретную начинку ПК на этапе компиляции и желание на все 100% использовать ресурсы оборудования. Кроме того, созданная «ручным» способом система получается очень компактной.

Ну и наконец, последняя, но вполне достаточная причина для установки *LFS* (хватило же некоторым ее для написания целого ядра операционной системы): «Just for Fun»!

Знакомство с LFS

Как рассказывает сам Герард Бикманс во введении, его знакомство с Linux состоялось в 1998 году. Перепробовав несколько дистрибутивов и не найдя ни одного, полностью отвечающего его требованиям, Герард решил собрать свой собственный вариант Linux из исходных текстов. Обобщив свой опыт в виде понятных и компактных инструкций и возглавив проект по поддержке этих знаний в актуальном состоянии, Герард вместе с единомышленниками создал *Linux From Scratch* в том виде, в котором этот проект существует и по сей день.

СОБЕРИ СВОЙ LINUX

Андрей Маркелов рассматривает дистрибутив для любознательных пользователей

Помимо книги с инструкциями, в рамках проекта распространяется загрузочный LiveCD со средой разработчика и собранный в одном месте архив исходных текстов, протестированных на взаимную совместимость компонентов Linux-системы со всеми необходимыми исправлениями. Рассмотрим же компоненты *LFS* подробнее.

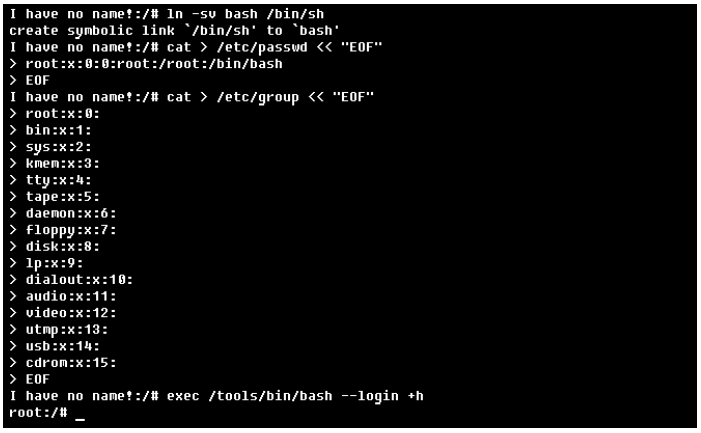
На момент написания статьи последней доступной автору версией *LFS* была 6.1.1.

Основная часть проекта – книга *Linux From Scratch* – содержит более 220 страниц и в семи главах описывает создание собственной Linux-системы, построенной на ядре 2.6.11 и включающей компилятор C/C++ gcc 3.4.3, язык программирования Perl 5.8.7, оболочку *bash 3.0* и основные GNU утилиты работы с системой, сетью, а также утилиты обработки текстов. Инструкции не включают описание сборки оконной системы X Window System и серверных служб.

Исходные тексты компонентов системы и исправления к ним занимают около 150 Мбайт. Помимо возможности скачать с сайтов разработчиков каждый пакет в отдельности, имеется возможность загрузить весь архив в составе Official *Linux From Scratch* LiveCD.

Официальный LiveCD (рис. 1, 2) – это не только архив всех исходных текстов и сама книга *LFS*, но и полностью готовая для сборки вашей *LFS*-системы среда, включающая *X Window System* и *Xfce*. Если вы скачали этот диск, вам, по сути, больше ничего не потребуется. Кроме всего прочего, диск включает в себя скрипты и файлы с инструкциями автоматической установки *LFS*. Данные средства предназначены для опытных пользователей *LFS*, которые хотя бы раз устанавливали систему «вручную». Новичку эти средства использовать не рекомендуется. Общий объем ISO-образа диска – 390 Мбайт.

Ну, и, наконец, последним в списке ингредиентов (но не последним по важности) *LFS*-коктейля будет хорошо приготовленный Linux-пользователь. (Я сказал «приготовленный»? Конечно, «подготовленный»!) Во-первых, если вы хотите использовать самую «свежую», англоязычную версию, вам необходимы минимальные знания технического английского для следования инструкциям книги. Существуют переводы и на русский язык, но они появляются с некоторой задержкой. Кроме того, большинство вспомогательных информационных ресурсов, ссылки на которые приведены статье, представлены только на английском языке. «Building an *LFS* system is not a simple task», – пишет Герард Бикманс. Безусловно, для того, чтобы собрать работоспособную систему Linux из исходных текстов, вы должны уметь работать с командной строкой, решать проблемы с компиляцией и иметь общее представление о том, как работает Linux.



```
I have no name!:/# ln -sv bash /bin/sh
create symbolic link `/bin/sh' to `bash'
I have no name!:/# cat > /etc/passwd << "EOF"
> root:x:0:0:root:/root:/bin/bash
> EOF
I have no name!:/# cat > /etc/group << "EOF"
> root:x:0:
> bin:x:1:
> sys:x:2:
> kmem:x:3:
> tty:x:4:
> tape:x:5:
> daemon:x:6:
> floppy:x:7:
> disk:x:8:
> lp:x:9:
> dialout:x:10:
> audio:x:11:
> video:x:12:
> utmp:x:13:
> usb:x:14:
> cdrom:x:15:
> EOF
I have no name!:/# exec /tools/bin/bash --login +h
root:/# _
```

Рис. 3.

Помимо самой книги *LFS*, на сайте проекта (**www.linuxfromscratch.org**) существует раздел часто задаваемых вопросов, список рассылки, также доступный по протоколу NNTP, IRC-канал и ряд информационных материалов. Кроме того, у вас будет под рукой документация, входящая в состав компилируемых пакетов, man-страницы, и, безусловно, множество ресурсов сети Интернет.

Также необходимо упомянуть, что кроме *LFS* на сайте **www.linuxfromscratch.org** «живут» несколько сопутствующих проектов: *Automated Linux From Scratch* (A*LFS*) – автоматизированная сборка из исходных текстов, Beyond *Linux From Scratch* (B*LFS*) – вопросы установки на уже готовую *LFS*-систему дополнительного программного обеспечения (X Window System, KDE, Gnome...), и другие.

«Ключ на Старт!»

Полная сборка *Linux From Scratch* на ноутбуке Sharp Actius MC24 (Mobile AMD Athlon XP-M Processor 2400+/512 MB RAM) с небольшими перерывами на домашние дела заняла одни выходные (Пользуясь случаем, хочу поблагодарить мою жену за терпение и понимание!). Для оценки затрачиваемого на сборку пакетов время, Герард вводит специальную единицу измерения – Standard Build Unit (SBU). Один SBU равен промежутку времени, затрачиваемому вашим ПК на выполнение команд

```
./configure...
make
make install
```

выполняющих сборку и установку самого первого из упоминаемых в руководстве пакетов – Binutil-2.15.

Далее по ходу изложения материала по книге даются примерные оценки времени затрачиваемого на пакеты в единицах SBU. Самые затратные с точки зрения времени компиляции пакеты: glibc-2.3.4 (12.3 SBU), gcc-3.4.3 (11.7 SBU), ядро Linux-2.6.11.12 (4.2 SBU) и Perl-5.8.7 (4.1 SBU). К слову сказать, один SBU на моем ноутбуке равняется приблизительно 4 мин. 23 сек. Сборка на том же оборудовании, но в среде *VMWare Workstation 5.0* под *Windows XP Service Pack 2* увеличила SBU более чем в два раза. Поэтому, если вы планируете уложиться в одни выходные – не используйте виртуальные машины.

Помимо сборки пакетов (кстати, некоторые из них собираются не по одному разу) время расходуется на различные **make check** и **make test**. Также прибавьте сюда время, которое вы потратите непосредственно на чтение книги и ввод команд.

Для сравнения, на страничке **http://www.linuxfromscratch.org/~bdubbs/** представлено более 150 конфигураций систем, предоставленных пользователями, собравшими *LFS*. Для каждой системы приведены значения SBU, так что вы всегда можете оценить время, необходимое для компиляции, исходя из данных для машин с аналогичной производительностью.

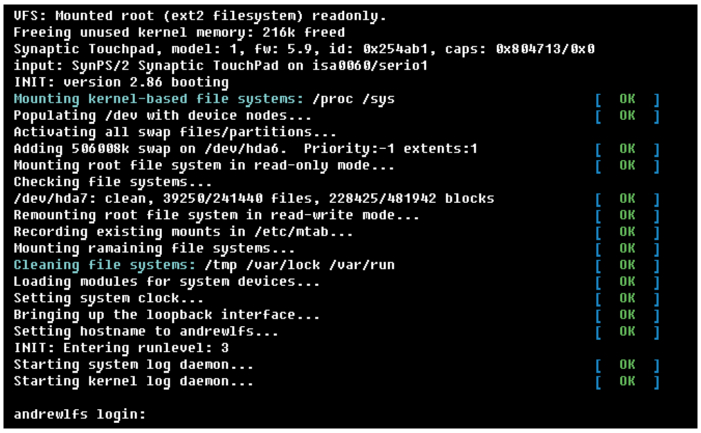
Этапы сборки собственной Linux-системы

Общий порядок сборки системы представлен в вводной главе и вкратце состоит из следующих действий.

Первым делом необходимо выделить для *LFS* собственный раздел на жестком диске, создать на нем файловую систему и подмонтировать к дереву файловой системы ОС, на которой будет производиться сборка. Далее, действуя из хост-системы (того же Official *Linux From Scratch* LiveCD), вы компилируете из исходных текстов среду разработчика, которую будете использовать для сборки *LFS*-системы. Затем, при помощи повторной пересборки GCC/Binutils и «песочницы» в виде chroot-окружения, *LFS* отделяется от хост-системы. После выполнения этого шага ваш *LFS* становится полностью самостоятельным, за исключением того, что он по-прежнему использует ядро хост-ситемы.

Далее, следуя подробным инструкциям, в «песочнице» компилируются все необходимые компоненты для нашей *LFS*-системы и по ходу сборки создаются необходимые конфигурационные файлы (рис. 3).

Наконец, в собранную систему добавляются стартовые скрипты, при необходимости, устанавливается и настраивается загрузчик, и, наконец, наступает долго- >>



```
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 216k freed
Synaptic Touchpad, model: 1, fw: 5.9, id: 0x254ab1, caps: 0x804713/0x0
input: SynPS/2 Synaptic TouchPad on isa0060/serio1
INIT: version 2.86 booting
Mounting kernel-based file systems: /proc /sys                    [  OK  ]
Populating /dev with device nodes...                              [  OK  ]
Activating all swap files/partitions...
Adding 506008k swap on /dev/hda6.  Priority:-1 extents:1          [  OK  ]
Mounting root file system in read-only mode...                    [  OK  ]
Checking file systems...
/dev/hda7: clean, 39250/241440 files, 228425/481942 blocks        [  OK  ]
Remounting root file system in read-write mode...                 [  OK  ]
Recording existing mounts in /etc/mtab...                         [  OK  ]
Mounting ramaining file systems...                                [  OK  ]
Cleaning file systems: /tmp /var/lock /var/run                    [  OK  ]
Loading modules for system devices...                             [  OK  ]
Setting system clock...                                           [  OK  ]
Bringing up the loopback interface...                             [  OK  ]
Setting hostname to andrewlfs...                                  [  OK  ]
INIT: Entering runlevel: 3
Starting system log daemon...                                     [  OK  ]
Starting kernel log daemon...                                     [  OK  ]

andrewlfs login:
```

Рис. 4.

жданный момент перезагрузки в систему, целиком собранную из исходных текстов. (рис. 4)

Поскольку *LFS* Book во многом ориентируется на людей, изучающих Linux, в каждом параграфе, посвященном сборке конкретного пакета, помимо инструкций по компиляции приведено примерное время, затрачиваемое на сборку, требуемое дисковое пространство, зависимости от других пакетов, и, наконец содержимое пакета с кратким описанием устанавливаемых утилит и библиотек.

Что дальше?

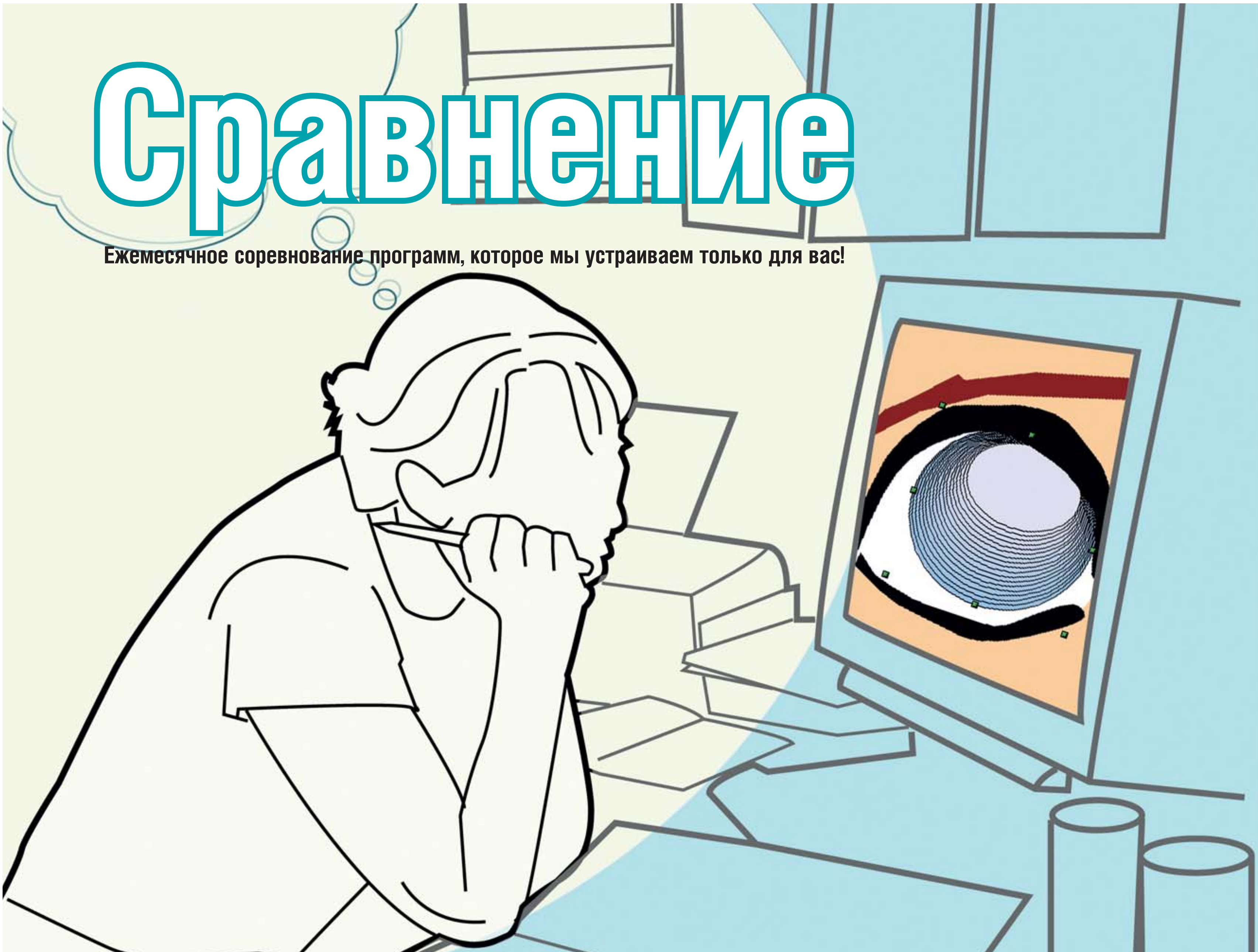
Безусловно, готовая *LFS*-система, собранная в соответствии с рекомендациями книги Герарда Бикманса, разительно контрастирует с современными Linux-дистрибутивами. В таком виде она мало подходит для решения реальных задач и повседневной работы, и способна лишь выполнять роль учебной системы для базового курса по Unix-системам, да и то в части, не касающейся настройки *X Window System*, которая просто не в ходит в список описанных в *LFS* пакетов. Чтобы довести *LFS* «до ума», в первую очередь советую вам обратиться к руководству *Beyond Linux From Scratch* (*BLFS*) доступному по адресу **http://www.linuxfromscratch.org/bLFS**. Данная книга построена по принципам и с использованием той же структуры, что и *Linux From Scratch*, но по размерам – раз в пять больше. В двадцати одной главе рассматриваются вопросы самостоятельной сборки более трех сотен пакетов (*X Window System*, Gnome, KDE), а кроме того – расширенная конфигурация системы и сборка дополнительных библиотек, вопросы компиляции клиентского и серверного программного обеспечения для работы в сети, мультимедиа, ПО для работы со сканерами и принтерами и многое другое.

Напоследок – несколько советов. Проверяйте и перепроверяйте введенные в процессе установки команды – если возникли ошибки компиляции, воспользуйтесь историей *bash*. Возможно, вы просто ошиблись во вводе. Не путайте предупреждения с ошибками, и читайте документацию – не все ошибки при проверках после сборки пакетов критичны. Раздел 1.4 книги *LFS* содержит алгоритм поиска ответов на вопросы, связанные с возможными проблемами. Придерживайтесь рекомендаций относительно того, как правильно задать вопрос и куда его направить, чтобы можно было расчитывать на помощь. Обратите внимание на раздел «What about netiquette?» в списке часто задаваемых вопросов. Не игнорируйте и FAQ в целом, а также список рассылки проекта. Возможно, вы не первый столкнулись с такой проблемой, и ответ уже можно найти в архиве списка рассылки. Обратите внимание на раздел **http://www.linuxfromscratch.org/hints/**, где приведены советы на темы, не освещенные в книгах *LFS* и B*LFS*. Прежде чем адресовать вопрос сообществу *LFS*, убедитесь, что вы использовали все доступные вам источники информации. Не забывайте и про форум нашего журнала (**www.linuxforum.ru**), а также рубрику «Вопрос? Ответ!». Направляйте свои письма на адрес **answers@linuxformat.ru**. Удачи и успехов в освоении увлекательного мира Open Source! **LXF**



Сравнение

Ежемесячное соревнование программ, которое мы устраиваем только для вас!



РЕДАКТОРЫ ВЕКТОРНОЙ ГРАФИКИ

Тебя ли с растром мне сравнить? Векторная графика на пике моды, поэтому **Ник Вейч** тестирует некоторые из лучших редакторов векторной графики с открытым исходным кодом: они годятся для дизайна, иллюстраций и просто для экраномарания.

Еще на заре графического пользовательского интерфейса (GUI) отдельные чудики запали на идею поиска оптимальных графических инструментов: сперва – чтобы создавать GUI собственных приложений, потом – чтобы использовать графический интерфейс как трамплин для творчества. Вся графика аккуратно поделена на две части: на растровую, где изображение состоит из пикселей, и векторную, или «структурированную» графику, тут объекты сохраняются в виде описаний формы, превращающихся в пиксели только во время отображения на экране или для печати.

Масштаб структурированной графики при необходимости можно легко варьировать, но ее применение зачастую сводилось к печатному дизайну. Благодаря появлению стандарта SVG (Scalable Vector Graphics, масштабируемая векторная графика) и системам рендеринга вроде Cairo, эту резервную роту вновь натренировали и отправили на передовую. А раз уж векторные графические редакторы Linux пошли в ход и для работы, и для развлечения, мы ощутили необходимость сравнить самые лучшие из них.

Наше исследование, опиравшееся на тестовый пакет W3C SVG Test Suite, фокусировалось на трех элементах. Во-первых, возможностях программ по созданию иллюстраций, от простого логотипа до полноценного произведения искусства. Такая работа требует изрядного количества манипуляций с объектами и кривыми, а также наличия продвинутых свойств, например, прозрачности. Другая область применения векторной графики – создание структурных диаграмм, схем и графиков. Способность поддерживать структуру при одновременном редактировании или перекомпоновке всего объекта здесь очень полезна. И, наконец, какой смысл создавать лучшие в мире графические шедевры, если их нельзя использовать, поэтому мы тщательно и строго рассматривали конечный результат и поддержку соответствующих стандартам файловых форматов.

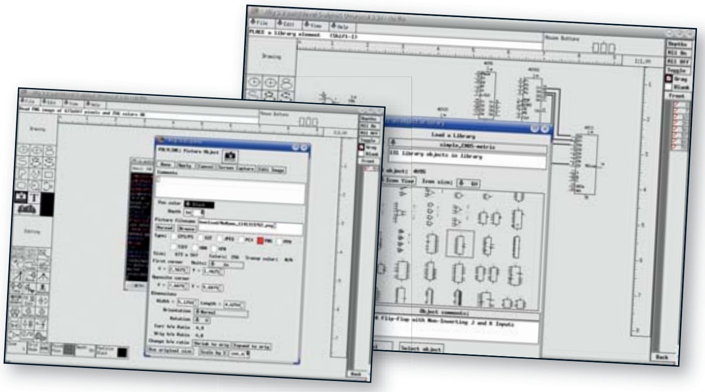
Как обычно, будут жалобы по поводу программ, не попавших в наше Сравнение. Позвольте сразу же на них ответить: да, *Xfig* вошел в наше Сравнение, потому что это приложение больше всех прочих напоминает живое ископаемое. Нет, *Xara* сюда не вошла, потому что до сих пор остается бета-версией, и включать ее было бы неправильно. Были еще разные инструменты на основе Java, которые либо ненадежно работали, либо не подвергались обновлению больше года. Если вы все еще думаете, что мы что-то пропустили – напишите вашу жалобу на листочке, привяжите его к кирпичу и бросьте в известное окно (адрес – в конце Сравнения).

УЧАСТНИКИ СРАВНЕНИЯ

Xfig НА ДИСКЕ

Приложение Gnome следующего поколения.

• **ВЕРСИЯ:** 3.2.5 •**WEB:** www.xfig.org • **Цена:** Бесплатно по лицензии GPL



Мы вовсе не утверждаем, что *Xfig* – старая программа, просто самый первый отчет о ее ошибках присылал еще Ной. Эта программа разрабатывается с незапамятных времен (на самом деле, с 1985 года), и в итоге дошла только до версии 3.2.5! Но пока ее охочие до ресурсов родственники слой за слоем наводят глянец, *Xfig* по-прежнему имеет что предложить.

Первый приятный сюрприз: родной файловый формат программы – FIG – настолько прост, что его можно с легкостью сгенерировать в другой программе или с помощью простых скриптов, и это укрепляет позиции Unix как части экосистемы других приложений, которые можно связывать друг с другом, отвечая сложным программным запросам. Во-вторых, хотя сам подход и не нов, но технологии остаются актуальными – например, *Xfig* поддерживает SVG и все шрифты, доступные X. Кроме того, весьма обширна поддержка *Xfig* третьими сторонами, особенно по части построения схем и графиков, с большим количеством библиотек доступных объектов.

Этот старинный, обнадеживающе-минималистичный инструмент Unix врос в хакерскую культуру и повсеместно остается любимым инструментом векторной графики у мужиков с большими кустистыми бородами. И не без причины. Может быть, *Xfig* и уступает более современным инструментам для иллюстраций, но по-прежнему высоко держит марку поддержки файловых форматов и взаимодействия с другими инструментами Unix.

***Xfig* поддерживает любую технологию, но истинно великим ему не бывать.**

ВЕРДИКТ LINUX FORMAT

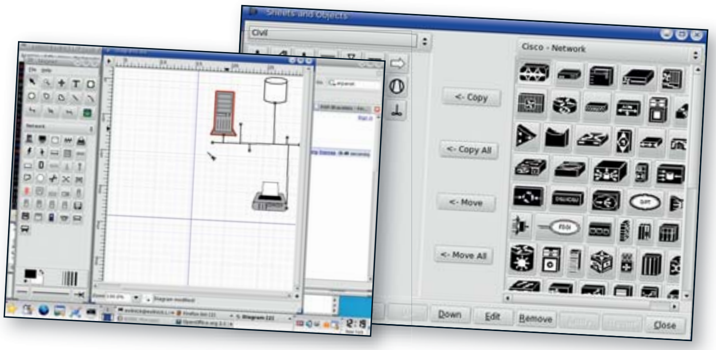
Не больно красив, но функционален и хорошо работает при минимуме требований.

РЕЙТИНГ 5/10

Dia НА ДИСКЕ

Присоединяйтесь – и вы получите очень полезный инструмент.

• **ВЕРСИЯ:** 0.94 • **WEB:** www.gnome.org/projects/dia • **Цена:** Бесплатно по лицензии GPL



***Dia* – основанный на GTK инструмент** для рисования, разработчики которого изначально хотели создать его похожим на *MS Visio*, инструмент для создания диаграмм, который без проблем работает со структурными элементами любого вида. Его хлеб с маслом – сетевые диаграммы, блок-схемы, документы и т.п.

Во всем, что не относится к вышеперечисленному, уровень его функциональности и эргономичности резко снижается. Через *GTK Dia* может использовать практически любой шрифт, но при этом не обладает никакими дополнительными инструментами форматирования текста (в отличие от, скажем, *Inkscape*), поэтому слова вводятся так же, как и любой другой объект, причем в одну строку, если вы не позаботитесь о переносах. Нет текстового блока, отсутствуют стили (за исключением поддерживаемых шрифтом), есть всего несколько опций выравнивания. Вы не найдете здесь возможности конвертировать текст в отдельные кривые, так что прощально помашите ручкой хитрым эффектам. Поскольку *Dia* нацелена на создание диаграмм, слабость текстовой поддержки в ней выглядит как-то предательски.

Современный GUI мог бы быть и поудобнее в использовании, чем у *Xfig*, но он использует те же самые разболтанные панели инструментов, которые засоряют *Gimp*, и в ответ на большинство действий всплывает диалоговое окно, только затрудняющее работу. Однако, когда дело доходит до любых схематических рисунков, на *Dia* можно и нужно рассчитывать.

Для диаграмм *Dia* хороша, а вот для рисунков не очень.

ВЕРДИКТ LINUX FORMAT

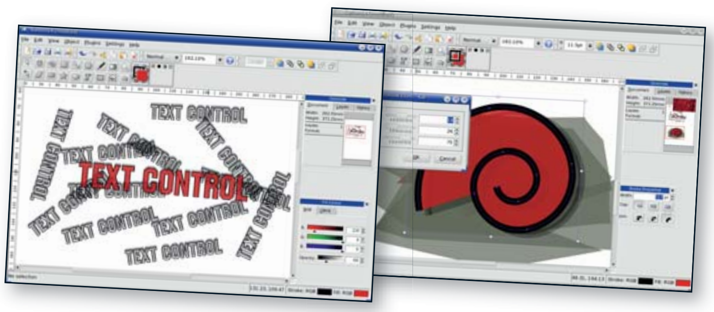
Не универсал, но как инструмент специалиста не хуже других.

РЕЙТИНГ 7/10

Karbon14 НА ДИСКЕ

Участник от KOffice.

• **ВЕРСИЯ:** 0.2 • **WEB:** www.koffice.org/karbon • **Цена:** Бесплатно по лицензии GPL



***Karbon14* – это самое последнее** название графического *К*омпонента из пакета *KOffice*: надоевший компании Adobe *KIllustrator* и незначительный *Kontour* благополучно отошли в историческое прошлое. Как это ни грустно, за период между названиями прогресса не просматривается.

Функции Импорта и Экспорта Графики рассчитывают на векторную графику, поэтому вам придется использовать инструмент Растровое изображение на панели инструментов, чтобы импортировать изображение в одном из трех поддерживаемых растровых форматов (PNG, JPEG и GIF). Панель инструментов нуждается в некоторой рационализации – наличие специальных кнопочек для изображения спирали или синусоиды, пусть и очень красивых, кажется излишним, учитывая, насколько редко их используют.

Приятно, что имеются эффекты тени (доступные из дополнительных модулей), но еще приятнее было бы иметь работающий рендеринг шрифтов, опции экспорта получше, шаг и повтор, и некоторые пункты в меню, которые не вызывали бы автоматический крах. Наверно, мы немного суровы к программе, ведь она вышла лишь в версии 0.2 и может считаться молодой-зеленой. Да к тому же *Karbon14* умеет читать и записывать формат SVG и OpenDocument (подвиг, с которым не справился *OpenOffice Draw*), а это чего-нибудь да стоит!

Функциональность *Karbon14* висит лоскутами, но все же многообещающа.

ВЕРДИКТ LINUX FORMAT

Местами блестящий, местами убогий; проклятие *KOffice* снова настигло его.

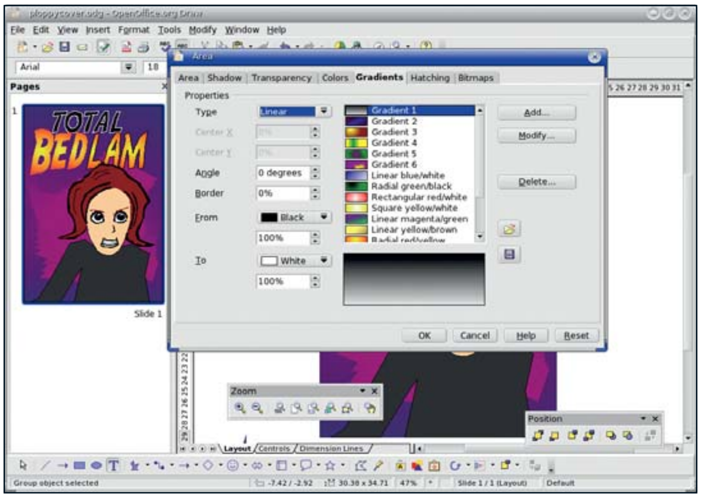
РЕЙТИНГ 5/10

OpenOffice.org Draw

НА ДИСКЕ

Да, тяжеленькая, но интеграция превосходна.

• **ВЕРСИЯ:** 2.0.1 • **WEB:** www.openoffice.org • **Цена:** Бесплатно по лицензии GPL



OOo Draw **отлично сочетает инструменты для создания грандиозных рисунков с генерацией эскиза одним щелчком.**

Если вы только что обновили *OpenOffice.org* 2.0, то могли заметить, что некоторые приложения радикально изменились. Особенно это верно насчет *Draw*, по крайней мере, внешне. Интерфейс перекроили, инструменты рисования стали больше и доступнее, а настраиваемые аспекты программы настолько расширились, что теперь запросто можно создать собственную панель управления и вынести туда инструменты и формы, которые вам нужны – по крайней мере, вы сумеете это сделать, прочитав документацию.

> «ИМЕТЬ ДОСТУП КО ВСЕМУ ОФИСНОМУ ПАКЕТУ ОЧЕНЬ ПОЛЕЗНО.»

Одна из странностей программы – вид по умолчанию: «страница» на левой панели и вместо текущего рисунка. Большинство художников привыкли работать над одним элементом, а не заполнять документы страницами. Единственным приемлемым объяснением этой странности может служить лишь то, что ныне все больше стираются грани между составными частями пакета *OOo*: документ затевается как чертеж, слайд-шоу или текст, а в итоге превращается в комбинацию из всех трех.

Это та область, в которой *Draw* имеет явно преимущество перед соперниками: за исключением Karbon, ни одна другая программа не в силах невзначай вставить ряд Фибоначчи посреди диаграммы, графика или полнофункциональной электронной таблицы. Для обеспечения комплексного взаимодействия разнородных материалов доступ к интегрированному офисному пакету, который может динамически соединять данные, весьма полезен.

Однако у офисного подхода есть и недостатки. Хороший пример – слои. Понятно, что графический редактор только выигрывает от наличия слоев, но как их применить наилучшим образом? В случае с *Draw*, они следуют той же модели, что и листы в документе *OOo Calc* – слои располагаются на вкладке внизу слева на панели просмотра, поэтому увидеть их разом нельзя. Зачем они нужны, тоже не совсем ясно: просто включать, выключать или блокировать слои – сомнительное удовольствие, особенно если учесть, что блокировка срабатывает не всегда, а на котором из слоев расположен элемент, нипочем не догадаешься.

Когда вы начинаете рисовать, *Draw* кажется вам почти что живым художником. Все стандартные инструменты на месте: кривые, квадраты, текст, простейшие формы, инструмент для рисования от руки… Есть даже такие красоты, за которыми погнался бы любой набор профессиональных инструментов: простая трассировка растров, привязка текста к контуру, эффекты шрифтов и смешивание.

Вишенка (слегка просроченная) на этом торте – 3D-эффекты, с помощью которых любой объект можно преобразовать в трехмерный и наиграться с ним всласть. Впечатляют рендеринг шрифтов и простые формы, которые, кажется, прямо слетают на экран. Всем этим следует пользоваться умеренно – как и клип-арт, поставляемый с программами, мудреные эффекты изрядно раздражают, если ими злоупотребить.

Носитель стандарта

По идее, *Draw* должен бы отлично работать с различными форматами файлов, ведь *OpenOffice.org 2.0* громко превозносили как первый серьезный офисный пакет, поддерживающий формат OpenDocument (нацеленный на будущее XML-стандарт, разработанный для взаимодействия с другими программами). Увы, поверхностная поддержка других форматов придает программе привкус тухлого яйца. Способность безвозмездно экспортировать ваши рисунки в проприетарный формат Flash (.swf) – это, конечно, замечательно, высший балл. Но вовсе не замечательна неспособность импортировать векторные изображения (SVG), хотя экспортировать в SVG легко и удобно.

Для *Draw* есть неофициальный дополнительный модуль, который использует движок *Apache Batik* для перевода файлов SVG в формат OpenDocument, но мы его здесь рассматривать не будем, поскольку он не входит в официальный релиз и полной поддержки стандарта SVG все равно не обеспечивает (заметно, впрочем, что разработчики постарались). Зато другие типы ввода *Draw* очень даже удались, включая ввод полезных, хоть и редко встречающихся файлов DXF (AutoCAD), EMF и RGB TIFF.

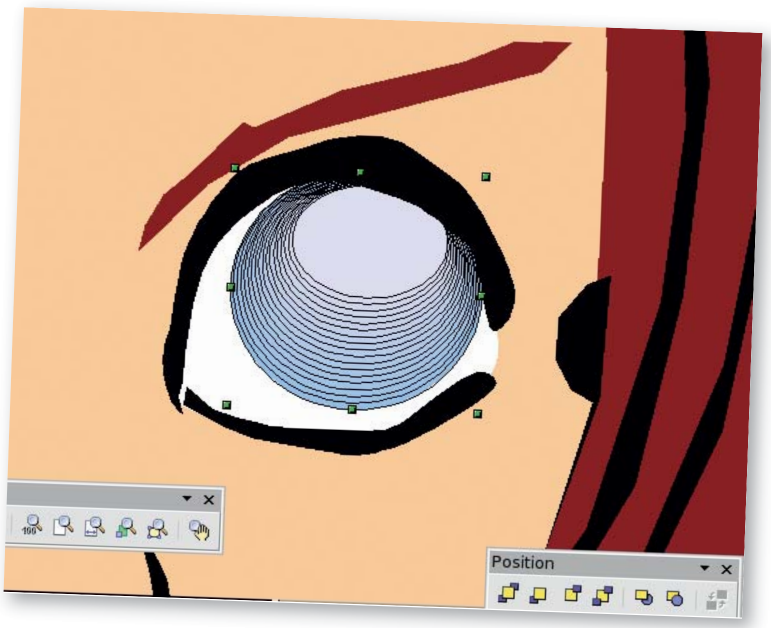
Более очевидный недостаток *Draw* – жадная до ресурсов натура всего пакета. Тонны чернил извели, обсуждая, сколько времени уходит на вызов *OOo* к жизни, но съедаемый им объем памяти почему-то обходят молчанием. А ведь если для редактирования растровых элементов вашего дизайна вы запустили и *Draw* и *Gimp*, проблемы обеспечены.

Какое место в нашем Сравнении занимает эта программа? Мы бы сказали, что в *Draw* есть все – по крайней мере, с виду. Стиль жизни у программы, может быть, и гламурный, но иногда она лежит ночью без сна и размышляет, не упустила ли чего-нибудь.

ВЕРДИКТ LINUX FORMAT

Многое делает очень хорошо, но дряблая, и местами ей нехватает воображения.

РЕЙТИНГ **8/10**



Для простых иллюстраций *OOo Draw* **более чем неплоха.**

Inkscape

НА ДИСКЕ

La belle du jour [царица бала, – прим. ред.] графических редакторов с открытым кодом.

• **ВЕРСИЯ:** 0.4.2 • **WEB:** www.inkscape.org • **Цена:** Бесплатно по лицензии LGPL

Некогда ответвившись от инициативы *Sodipodi*, *Inkscape* сейчас лидирует среди инструментов векторной графики для Gnome. Хотя разработчики OpenOffice.org заявляют, что SVG был задуман как формат вывода (истинная правда), авторы данного кода этого в голову не брали. *Inkscape* использует SVG в качестве «родного» формата, и, по большому счету, это хорошо.

Приходится признать, что *Inkscape* не на 100% выполняет тестовые требования SVG 1.1, но ведь и все прочие программы из тех, что мы испытывали, тоже их не выполняют – в частности, дополнения к рендерингу текста и режимы наложения работают в режиме ограниченной функциональности (проще говоря, не поддерживаются). Однако же это не мешает редактировать документы, в которых используются эти свойства, или выводить их на экран.

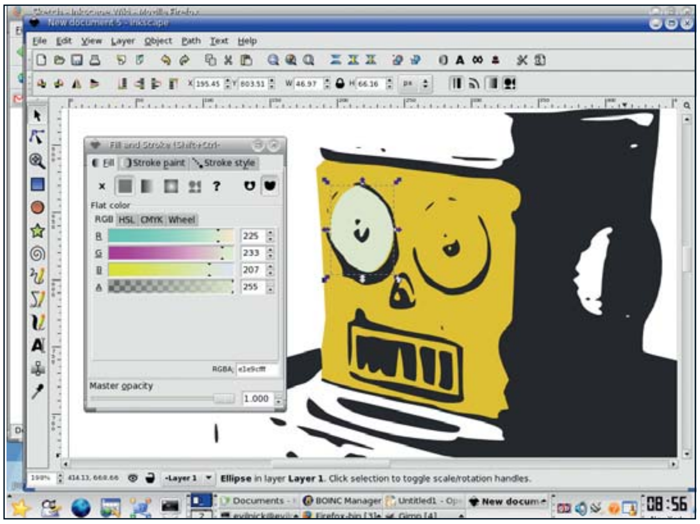
> «INKSCAPE ДАЕТ БОЛЬШЕ СВОБОДЫ – НО БОЛЬШЕ ТРЕБУЕТ ОТ ПОЛЬЗОВАТЕЛЯ.»

Минимализм = голяк

С точки зрения интерфейса, *Inkscape* выглядит очень чисто – едва ли не слишком чисто. Инструменты рисования прибились к левой стороне окна, а остальные элементы управления и прочая информация находятся в двухэтажной панели инструментов наверху. Любая операция посложнее вызывает уместное (в большинстве случаев) диалоговое окно.

Функция Свойства Объекта (Object Properties) – первая заповедь поддержки SVG данной программой, но выполняется она странновато. Щелкните правой кнопкой мыши, чтобы получить всплывающее меню и выбрать Свойства Объекта. Возможно, вы ожидаете увидеть информацию о цвете объекта или толщине штриха (зря ожидаете, эти данные – в пункте меню Заливка) или, по крайней мере, расположении объекта на странице (тоже зря, оно показано в панели инструментов). Все, что вы видите – это диалог, позволяющий редактировать XML-свойства объекта. Круто, конечно, что вы можете редактировать на таком уровне – это крайне важно, но способ исполнения неоправданно запутан. В большинстве случаев пользователь пребывает в блаженном неведении нюансов, отличающих Id, Label и Title, а именно эти пункты вы первым делом и увидите.

Работа со слоями тоже огорчает, хотя и не до такой степени, как в *Draw*. На сей раз слои представлены в виде выпадающего списка внизу страницы. Нельзя отрицать, что здесь хватает возможностей для полезных действий над слоями, но различать слои при работе трудно. Здесь и только здесь можно было бы позаимствовать принципы пользовательского интерфейса из *Gimp* – но не сообразили.

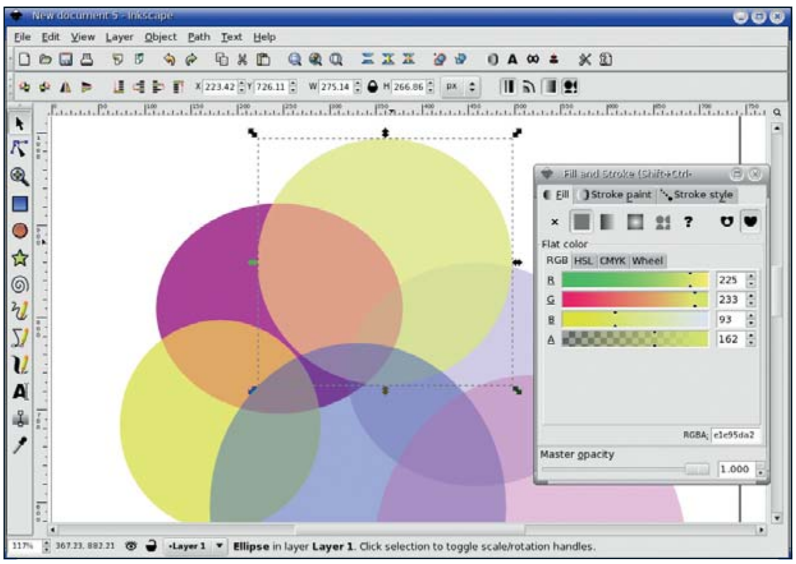


В *Inkscape* всего лучше удается трассировка растровых изображений. И раскраска роботов.

Свободное рисование

Но все это мелочи по сравнению с богатым выбором инструментов и простотой их использования. Программа, например, проще всех создает градиенты. Трассировка растровых изображений дала наилучшие результаты из всех инструментов, рассмотренных в этом Сравнении (причина, скорее всего, в возможности более тонкой настройки), а уж с текстом можно делать потрясающие вещи. Там, где *Draw* имеет тенденцию склоняться к предустановленным эффектам, *Inkscape* дает больше свободы, хотя и требует от пользователя куда больше взамен.

Поддержка файловых форматов в *Inkscape* достойная. Растровые форматы включают TIFF и Targa, но, вероятно, еще полезнее структурная поддержка таких форматов, как *Adobe Illustrator*. Вывод немного более ограничен: можно выбирать из растрового вывода, небольшой группы форматов *PostScript* (включая PDF) и *Inkscape* или простой SVG (сжатый или нет, в обоих случаях). Еще имеется поддержка довольно странного экспорта в POV-Ray, но его мы тестировать не стали.



Не все инструменты Linux поддерживают прозрачность – *Inkscape* и поддерживает, и экспортирует в большинство форматов, даже в SVG!

Разработчики постарались включить сюда и некоторые полезные для хакеров инструменты. *Inkscape* поддерживает скрипты, созданные с помощью *Python* – но чего ради вылезать на консоль и запускать Vi, если можно посмотреть скрипт изнутри? Более того, встроенный редактор XML позволит знающему пользователю проникнуть внутрь структуры рисунка и сделать быстрые исправления или добавить дополнительные данные в файл.

Из всех пакетов, именно в этом проще всего рисовать сложные формы. Инструменты рисования легко выбирать, и они выполняют то, что вы от них хотите; возможность применять многочисленные слои и прозрачность, особенно при смешивании, позволяют ускорить создание сложного изображения. У предыдущих версий *Inkscape* были проблемы с некоторыми аспектами прозрачности – то, что хорошо выглядело в *Inkscape*, могло оказаться серым в браузере или в другом приложении SVG. Теперь эти погрешности ликвидированы. Отсутствие должных смешений объектов отбрасывает программу назад, но так как в настоящий момент это один из наиболее активно развивающихся векторных инструментов, надо думать, что недостатки будут устранены.

ВЕРДИКТ LINUX FORMAT

Отсутствует несколько важных функций, но в остальном – очень, очень впечатляет.

РЕЙТИНГ 8/10

Sketsa

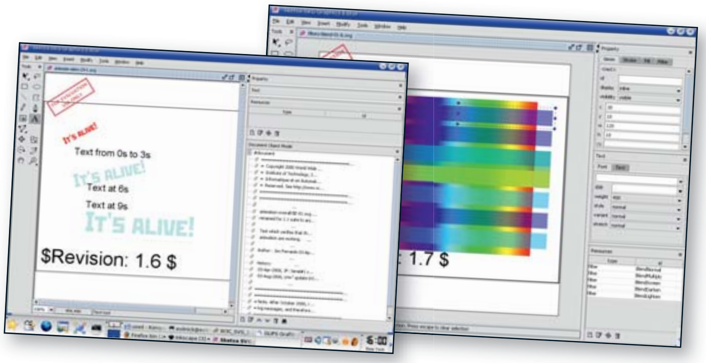
Аскетичная коммерческая попытка.

• **ВЕРСИЯ:** 3.3 • **WEB:** www.kiyut.com • **Цена:** $49

***Sketsa* отличается от инструментов,** которые мы рассматривали, по двум причинам: она использует Java, и она создана исключительно как SVG-редактор. Последнее никак не должно стать препятствием ни для вдохновения, ни для успеха этой программы. В конце концов, SVG создавался, чтобы быть *lingua franca* векторной графики, так что довольно скоро все подобные приложения станут редакторами SVG.

Однако *Sketsa* применяет спартанский подход – ничего лишнего, как будто функциональность и дружественность по отношению к пользователю – это экстравагантное украшательство, от которого нужно избавиться любой ценой. Интерфейс *Swing* не производит приятного впечатления, функции разбросанные, и хотя *Sketsa*, возможно, на данный момент лучшее приложение для рендеринга SVG-файлов, в самом приложении такие сложные изображения не создать. Более того, панели, выполненные в «инспекторском» стиле с правой стороны, производят впечатление, что они недалеко ушли от самого XML – хотя это может кому-то и понравиться.

Вы, очевидно, рассчитываете, что *Sketsa* сможет безупречно читать и записывать SVG файлы, и действительно, она хорошо справилась с тестовыми изображениями. Но, к сожалению, вы вряд ли наметаете здесь изысканный шедевр – в программе нет ни смешивания, ни слоев.



Хорошая поддержка SVG, а в остальном – полная ерунда.

ВЕРДИКТ LINUX FORMAT

Хорошая поддержка формата SVG, хороший рендеринг, а сверх этого почти ничего.

РЕЙТИНГ 5/10

Glips Graffiti

НА ДИСКЕ

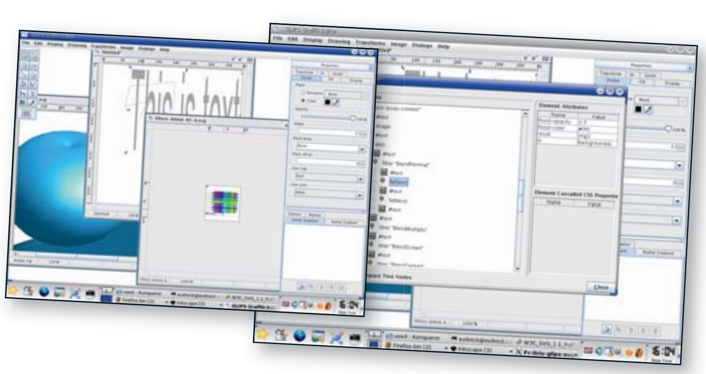
Кросс-платформный редактор на основе Batik.

• **ВЕРСИЯ:** 1.3 • **WEB:** http://glipssvgeditor.sourceforge.net
• **Цена:** Бесплатно по лицензии Apache

Сходство между *Glips Graffiti* и *Sketsa* заметно сразу. Обе программы используют Java, и ни одна из них не претендует на большее, чем роль просто хорошего редактора. Однако опыт работы с ними в качестве пользователя не позволяет продолжить аналогию. Хотя *Graffiti* выглядит малость симпатичнее, ей, мягко говоря, не хватает отзывчивости. Хорошо еще, что тут нет сложных процедур вроде смешения или прозрачности – они бы вообще все затормозили.

Очень жаль – хотя бы потому, что бесплатная программа хуже своего ближайшего конкурента. В ней тоже не хватает любви, да еще и функций маловато. Понятно, что *Glips Graffiti* пока что в развитии, но от версии 1.3 вы вправе ожидать большего. Минимальная поддержка файловых форматов и никаких функций, кроме рисования простых форм... наш поиск программы-победителя значительно сужается.

Хотя программа, безусловно, работает, и провела впечатляющий рендеринг наших SVG, все же на нее не захочется тратить время, чтоб создать рисунок с нуля. Пока не доросла.



Бедная родственница *Sketsa*. Увы.

ВЕРДИКТ LINUX FORMAT

Единственное, за что можно похвалить программу – хороший рендеринг SVG.

РЕЙТИНГ 3/10

Skencil

НА ДИСКЕ

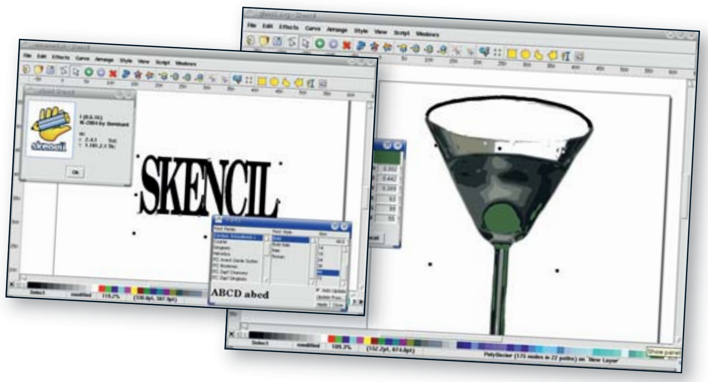
Инструмент для создания зарисовок на Python.

• **ВЕРСИЯ:** 0.6.16 • **WEB:** www.nongnu.org/skencil • **Цена:** Бесплатно по лицензии GPL

Эту маленькую жемчужину создал Бернхард Херцог (Bernhard Herzog) на *Python* и *Tkinter*, интерфейсе *Python* для *Tk* – инструментария разработки графических интерфейсов. Не ждите продвинутых функций – при нашем тестировании *Skencil* упорно сопротивлялся всем попыткам заставить работать смешивание. Хотя рендеринг шрифтов действительно хорош, в плане достижимых результатов он все еще ограничен.

Это инструмент для набросков [оно и логично, судя по названию, – прим.ред.], то есть он лучше всего приспособлен для того, чтобы начеркать что-нибудь, а вовсе не для создания подробной схемы ротора, и, как большинство таких инструментов, не предлагает функции соединения объектов наподобие имеющейся в *Dia*. Зато есть простой и чуткий интерфейс с понятными иконками, позволяющий вам в шесть секунд намалевать что-то в стиле если не Айвазовского, то хотя бы Пушкина (на полях). В нем нет прозрачности, блендинга, трассировки растрового изображения и многого другого... но, минуточку, что это? Слои, перечисленные, как положено, списком на определенной панели, в стиле, похожем на *Gimp*! Вы действительно можете увидеть, какие слои активны! И даже сказать, где находятся объекты! Вот и думай тут...

Касательно файлов, *Skencil* поддерживает SVG, WMF, PDF и Adobe Illustrator для экспорта, хотя при считывании этих форматов наблюдалась некие проблемы. Но все же как быстрое, использующее скрипты маленькое приложение для зарисовок оно имеет свои достоинства.



Лихо рисует, использует *Python*.

ВЕРДИКТ LINUX FORMAT

Простой, но эргономичный и понятный интерфейс. Результат зависит от формата файлов.

РЕЙТИНГ 5/10

РЕДАКТОРЫ ВЕКТОРНОЙ ГРАФИКИ ВЕРДИКТ

Было бы здорово, если бы всегда существовал «чистый» победитель нашего состязания – но если бы все было настолько очевидно, вы бы нашли его и без нас, и не нужны были бы долгие дни кропотливых исследований, которые привели к обнародованию этих результатов!

В данном случае победителя выбрать трудно, потому что большинство приложений, рассмотренных здесь, имеют свои козыри. Например, в создании структурных схем трудно превзойти *Dia*, а *OOo Draw* выигрывает благодаря своей возможности взаимодействия с другими элементами пакета *OOo*.

Однако *OOo Draw* не может импортировать SVG без дополнительного модуля, поэтому за ним остается второе место после *Inkscape*. *Inkscape* быстр, прост в использовании, имеет кучу функций, и более чем способен создавать высококлассные иллюстрации или простые для понимания схемы. Его интерфейс прост и эргономичен, хотя способ представления слоев стоило бы еще продумать. *Inkscape* все еще очень активно перерабатывается, и в этом, возможно, его главный плюс.

Сравните его со узкоспециализированным редактором SVG, например, *Sketsa*, и огромная разница в амбициях станет очевидна.

Мир векторной графики способен на сюрпризы. Учитывая продолжающиеся попытки портировать *Xara* в Linux (как Open Source), интересно будет посмотреть, во что выльются эти приложения через полгода. LXF

INKSCAPE 8/10

СРАВНИТЕЛЬНАЯ ТАБЛИЦА

	Inkscape	OOo Draw	Xfig	Dia	Karbon14	Glips Graffiti	Skencil	Sketsa
Версия	0.42	2.0.1	3.2.5	0.94	0.2	1.3	0.6.16	3.3
Лицензия	GPL	LGPL	Custom	GPL	GPL	Apache	GPL	Proprietary
Цена	Free	Free	Free	Free	Free	Free	Free	$49
Инструментарий	GTK	GTK	X	GTK	KDE	Java	Python/Tk	Java
Платформы	Linux, Win, Mac	Linux, Win, Mac	Linux, Mac**	Linux	KDE	Java	Linux, Mac	Java
Поддержка шрифтов	All	All	Latex/PS	All	All	All	All	All
Перевод текста в кривые	✓	✓						✓
Привязка текста к контуру	✓	✓			✓****		✓******	
Прозрачность	✓	✓			✓	✓		✓
Смешивание/tweening	*	✓					✓	
Клонирование	✓							
Связывание объектов	✓	✓		✓				
Трансформации	✓	✓						✓
Библиотеки			✓	✓				
3D-эффекты		✓						
SVG XML редактирование	✓							✓
Слои	✓	✓	✓	✓	✓		✓	
Трассировка растра	✓	✓						
Язык макросов	Python	StarBasic		Python	Plugins		Python	
Input Входные форматы								
SVG	✓	*****		✓***	✓	✓	✓***	✓
WMF	✓	✓			✓		✓	
Adobe	✓				✓		✓	
Dia	✓			✓				
AutoCAD DXF		✓						
Xfig			✓	✓			✓	
Форматы вывода								
EPS	✓		✓	✓	✓			
Tex/Latex	✓		✓	✓				
PDF	✓	✓	✓		✓		✓	
SVG	✓	✓	✓	✓	✓	✓	✓	✓
Bitmap	✓	✓	✓	✓	✓	✓	*****	✓
Flash (.swf)		✓						
Adobe							✓	
WMF		✓			✓		✓	

* с помощью экспериментального модуля ** порт для Windows – *Winfig* *** неполная совместимость **** только трансформируется ***** существует plug-in ****** только к ломаной кривой

Hot Picks

НА ДИСКЕ

Лучшее на планете новинки открытого ПО!



Майк Сондерс
Прочитав 1984, Майк решил тестировать все ПО в комнате 101.

Сотни программ с открытым кодом создаются и обновляются каждый месяц, поэтому среди них непросто найти достойную использования. В разделе HotPicks мы просеиваем все последние версии различных программ и отделяем зёрна от плевел, отбирая только самое лучшее. И ежемесячно самое выдающееся приложение получает наш приз.

Вы можете опробовать любую из программ обзора, установив её с прилагаемого к журналу диска. Если вы заметили программу, которую стоит рассмотреть, черкните письмо на **mike.saunders@futurenet.co.uk**.

HOTPICKS В ЭТОМ РАЗДЕЛЕ

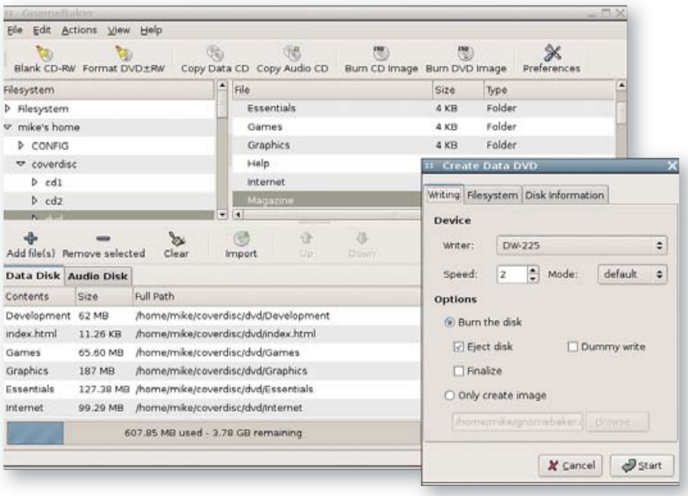
БРАТИТЕ ВНИМАНИЕ НА ПОБЕДИТЕЛЯ HOTPICKS

Все, что попадает в раздел **HotPicks**, заслуживает самого пристального внимания. Однако каждый месяц мы выбираем всего один самый яркий проект. Побеждает лучший!

ПРОГРАММА ДЛЯ ЗАПИСИ CD И DVD

GnomeBaker

• **ВЕРСИЯ** 0.5.1 • **WEB** http://gnomebaker.sourceforge.net/v2



***GnomeBaker*: немного не хватает силенок, чтобы помочь производству грандиозных дисков для журнала *LinuxFormat*, но до этого недалеко.**

Сегодня чуть ли не каждый компьютер оборудован пишущим CD-приводом, а многие машины класса high-end – еще и пишущим DVD-приводом. Смешная стоимость CD- и DVD-болванок плюс необходимость постоянного создания резервных копий систем и данных обуславливают жизненную необходимость программного обеспечения для записи оптических дисков – только поглядите на огромный рынок Windows-приложений. Здесь, в мире Linux, тоже есть выбор, от *K3b* до программ, управляемыми из командной строки; но всегда есть место для других подходов. *GnomeBaker* был на диске, прилагавшемся к журналу за предыдущий месяц; он достоин более пристального внимания, это серьёзное приложение для записи дисков.

По названию *GnomeBaker* понятно, что для его компиляции из исходных текстов потребуются библиотеки Gnome. Само приложение диски записывать не умеет; соответствующую функциональность предоставляет вездесущий набор программ *Cdrtools*: утилиты командной строки для создания образов, записи образов на диск, извлечения аудиоинформации с компакт-дисков, и т.д. В вашем дистрибутиве, вероятно, уже есть *Cdrtools*, а если нет, берите его с диска, прилагаемого к журналу.

> «УХИТРЯЕТСЯ ДОСТИЧЬ ТОГО, ЧТО НЕ ПОД СИЛУ ДРУГИМ ГРАФИЧЕСКИМ ПРИЛОЖЕНИЯМ.»

Интерфейс *GnomeBaker* сразу подводит к делу: большие иконки расположены в верху основного окна, под ними находится масштабируемый графический элемент для просмотра файловой системы. Ещё пониже – вкладка, на которой перечислены файлы и каталоги, предназначенные для записи на диск, с указанием их размера. (Вкладку можно переключить на просмотр аудиодорожек для записи музыкальных дисков.)

Процесс записи дисков максимально упрощён. Достаточно просто перетащить файлы из навигационного окна вниз, а *GnomeBaker* доделает остальное. Поддерживаются разные типы дисков, от «визитки» на 200 МБ до двустороннего DVD ёмкостью 8.5 ГБ.

Запись

Когда вы выбираете тип диска, внизу появляется панель, показывающая, сколько осталось свободного места. В диалоговом окне записи дисков можно выбрать значения по умолчанию, а также расширений Rock Ridge/Joliet и скорость записи, и заполнить информационные поля, например, имя издателя. Если у вас уже есть ISO-образ диска, вы можете записать его парой щелчков мыши; программа также поддерживает непосредственное копирование данных и музыки с дисков. Единственный аспект, который может запутать новичков – это выбор устройства; к счастью, *GnomeBaker* включает функцию автоматического поиска и определения устройств.

Основной недостаток *GnomeBaker* – это отсутствие возможности сохранения раскладки: проще говоря, нельзя запомнить структуру диска для дальнейшего использования. Если вы всего лишь скидываете на компакт-диск резервную копию программного обеспечения, это не беда; но если вы работаете над довольно сложной структурой DVD, вам, естественно, хотелось бы сохранить результаты вашей работы для нового употребления.

Не помешала бы ещё парочка кое-каких мелких функций, например, проверка контрольной суммы MD5 для верификации записи, но в остальном это весьма тщательно продуманное ПО. *GnomeBaker* ухитряется продемонстрировать то, чего не смогли добиться разработчики многих других GUI-программ: идеального соотношения между простотой использования и дополнительными опциями для требовательных пользователей. Версия 1.0 обещает быть шедевром.

ПЕРСОНАЛЬНЫЙ ФИНАНСОВЫЙ МЕНЕДЖЕР

Opale

• **ВЕРСИЯ** 0.7.1 • **WEB** http://orzel.freehackers.org/opale

Канули в Лету те дни, когда брать верх над финансовыми документами означало запихивать пачку бумаг себе под матрас. При столь огромном количестве банков, счетов и интернет-магазинов, борющихся за наше внимание, за расходами и не уследишь. Персональные финансовые менеджеры типа *GnuCash* и *Grisbi* пытаются решить эту проблему, но многие считают их слишком сложными или накрученными. Томас Капричелли (Thomas Capricelli) хотел, чтобы его новое приложение *Opale* было простым и непосредственным и его пользователь мог получать жизненно важную финансовую информацию без лишних заморочек.

Opale – приложение KDE, поэтому для компиляции данной программы вам понадобятся библиотеки разработчика для среды KDE и Qt (обычно называемые qt-devel и kde-devel в большинстве дистрибутивов Linux).

При первом запуске *Opale* не выдаст дружелюбное окно приветствия и не запустит никакого помощника: на экран выводится голый интерфейс. В нём вы и можете вводить исходящие платежи (на данном этапе *Opale* работает только с расходами, но – чего скрывать! – проблемы у большинства из нас скорее по этой части). Благодаря панели инструментов и строке состояния интерфейс довольно понятный, и раскопки в документации не понадобятся; правда, он мог бы быть и поприветливее.

Первый шаг – создание нового файла с названием банка и выбор валюты (все данные сохраняются в XML). Затем можно добавить индивидуальные исходящие платежи и разместить их по конфигурируемым пользователем категориям, например, Personal и Work, с указанием типа платежа (чек или наличные). После ввода комментариев и денежных сумм вы увидите записи в обратном хронологическом порядке и итоговую сумму – подлежащую выплате?! Ой…

Можно пометить некоторые платежи, временно исключив их из итога, и составить простенький график для наглядной оценки вашей финансовой ситуации. Документации не так чтобы много, только небольшое руководство в формате PDF, но очень легко обойтись и без него. Как свойственно версиям с промежуточным номером, хватает и недоработок, и непонятных глюков, но *Opale* присущ разумный баланс между необходимой функциональностью и простотой, и если вам нужна программа для отслеживания ваших финансовых потоков, над которой не надо ломать голову, дайте *Opale* шанс.

«ДАСТ ВАМ НЕОБХОДИМУЮ ФИНАНСОВУЮ ИНФОРМАЦИЮ БЕЗ ЛИШНИХ ЗАМОРОЧЕК.»



Присвойте каждому платежу дату, *Opale* ее запомнит и сможет предупредить вас, что вы угодили в должники.

ФАЙЛОВЫЙ МЕНЕДЖЕР

PCManFM

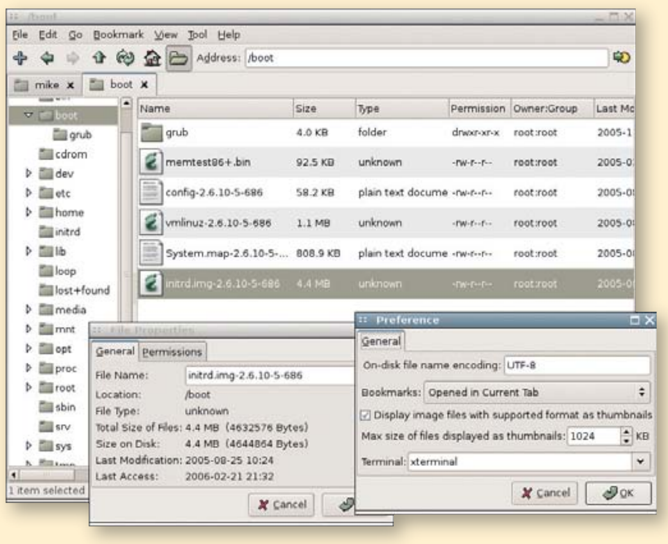
• **ВЕРСИЯ** 0.2.1 • **WEB** http://pcmanfm.sourceforge.net

***Konqueror* для KDE и *Nautilus* для** Gnome делают замечательную работу. Собрав предварительный просмотр, доступ в сеть и доступ к файлам в одно окно, они подняли процесс управления файлами на новый уровень. В итоге – это не самые легковесные приложения, и если пользователю нужно всего лишь попростому распихивать файлы, получается перебор. PCMan File Manager идеально подходит тем, кто хочет без затей привести свои файлы в порядок; данная программа может похвастаться высокой скоростью и дружественным интерфейсом.

Для её компиляции нужна *GTK* версии от 2.8 и *Gamin*, чтобы отслеживать изменения в файлах. Визуальная часть приложения потребует **gnome-icon-theme**, но *PCManFM* не опирается ни на какие библиотеки или программы среды Gnome. После компиляции вы получите файлик размером 300K, использующий всего 8 МБ оперативной памяти. Может, это и не ультралёгкое приложение, но для старых компьютеров явно лучше, чем *Nautilus* и *Konqueror*, которым подавай 15 и 17 МБ соответственно.

Теперь поскорее скажем: «Без паники!» Вы, кто презирает пространственный режим *Nautilus*'а и скривил гримасу при виде картинки справа от текста, знайте: *PCManFM* выводит все сведения в одно окно. При желании можно даже вызвать дерево каталогов а-ля *Windows Explorer* или просмотреть более подробную информацию о файле в главном окне. Присутствуют все полагающиеся функции: сортировка, удаление и копирование файлов перетаскиванием. *PCManFM*, конечно, не образец державной мощи, но доступ ко всему вы можете получить, через панель инструментов и контекстные меню, со скоростью молнии.

Несмотря на легкий вес, *PCManFM* снабжен такими удобствами, как просмотр со вкладками и возможность открыть новое окно терминала прямо из текущего каталога. Также имеется простая система закладок. Можно бы добавить кое-какие функции, причем без повышения требований к оперативной памяти – например, сортировку по колонкам в окне списка и автоматическое завершение адресной строки. Но вообще-то *PCManFM* – опрятная и приятная в использовании программа. А главное, она уже есть, поэтому, если вы используете легковесный менеджер окон или *Xfce* и не хотите дожидаться запуска *Thunar*, данная программа удачно пополнит ваш рабочий стол.



Высокоскоростной *PCManFM* – естественное дополнение к *Xfce*, но прекрасно работает и с любым другим оконным менеджером.

КОМАНДНЫЙ ИНТЕРПРЕТАТОР

Fish

• **ВЕРСИЯ** 1.20.2 • **WEB** http://roo.no-ip.org/fish



IMAGE COURTESY: AXEL LILJENCRANTZ, 2005

Интерпретатор командной строки Linux часто называют самым мощным и многоцелевым элементом операционной системы. Он позволяет проделывать такие вещи, о которых нечего было и думать, работая с графическим интерфейсом. Комбинируя различные команды и преобразуя их выход, вы можете, например, написать скрипт, за секунды выполняющий переименование или масштабирование сотен картинок определённой ширины, чтобы вам не пришлось часами трудолюбиво терзать для этого мышь. И все же многие пользователи, включая ветеранов компьютера, находят неудобной лаконичность родной оболочки Linux (*Bash*).

Fish покушается всё изменить. Данный командный интерпретатор включает функции, способные облегчить жизнь как новичкам в мире Linux, так и опытным пользователям, сохраняя всю мощь командной строки и одновременно уводя её от интерфейса, недоступного интуиции. Чтобы запустить *Fish*, можно установить **Fish-1.20.2-0.i386.rpm** с диска этого месяца или пойти по пути компиляции из исходных текстов. В последнем случае понадобится всего-навсего набрать (от имени суперпользователя) команды **configure**, **make** и **make install** в окне терминала, а потом набрать **Fish** для запуска программы. Экзотических зависимостей нет; уже хорошо.

После запуска *Fish* вы первым делом замечаете раскрашенную подсказку командной строки. В принципе, и *Bash*, и другие интерпретаторы можно настроить на использование цветов, но по умолчанию это редкость. *Fish* распоряжается цветами со вкусом: зелёный отделяет текущий каталог от имени пользователя и названия хоста; это мягкая форма подсветки синтаксиса.

> «FISH ПРЕДЛАГАЕТ ВПОЛНЕ УМЕСТНУЮ МОДЕРНИЗАЦИЮ СИНТАКСИСА СКРИПТОВ.»

При вводе команды буквы сначала красные, а когда компьютер распознает нечто осмысленное, они окрашиваются в зелёный цвет. Например, «kons» густо краснеет, пока вы не введёте слово полностью («konsole»). Пустячок, а приятно.

Как и большинство современных оболочек, *Fish* позволяет дополнять команды клавишей «Tab», т.е. вы вводите первые несколько букв из названия команды или имени файла, нажимаете «Tab», и *Fish* дописывает команду за вас. Но и это ещё не всё! Используя страницы руководства, которые хранятся в системе, интерпретатор может отобразить возможные опции команды. Например, если вы наберёте **ls --** (двойной дефис) и нажмёте «Tab», *Fish* выведет на экран все доступные ключи команды *ls*, например, **--all**. Они тоже выделяются разными цветами, так что если вы забыли нужную вам опцию и ленитесь листать серые страницы руководства, это очень экономит время.

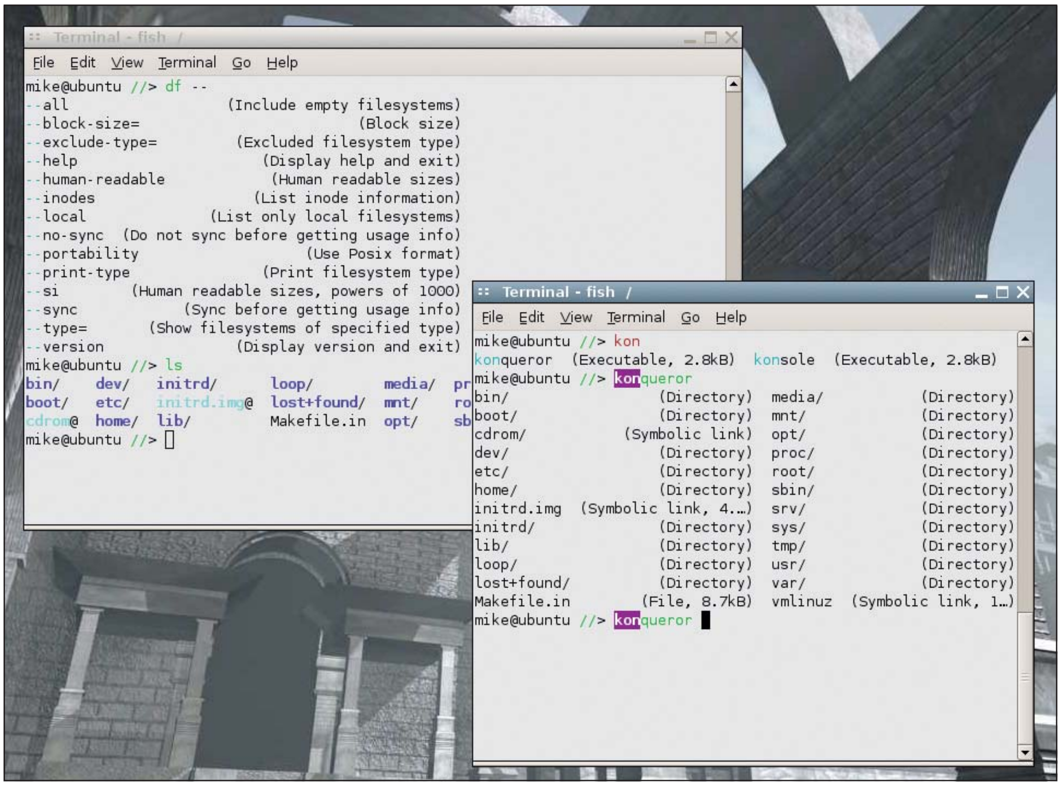
Фишки Fish

Аналогично, вы можете вызвать команду из стека предыстории команд (ибо он предусмотрен), набрав первые буквы команды и нажимая клавишу «Вверх». Т.е. если вы час-другой назад ввели длинную команду, начинающуюся с *df*, то можете просто набрать **df** и нажимать стрелку вверх, для поиска – нет нужды пролистывать весь список команд. Эта функция полезна не только храбрым новичкам, делающим первые шаги в командной строке, но и многолетним пользователям Unix. Вдвойне чудесно.

Копая глубже, вы обнаружите, что *Fish* включает также некоторые уместные усовершенствования синтаксиса языка сценариев. В традиционных скриптах наблюдается изрядный разнотык в структурах, например, *if* заканчивается на **fi**, for – на **done**, а *Fish* вносит во всё это логику и единообразие: в *Fish* закрывающая операторная скобка всегда **end**, и нету пестрой смеси дескрипторов окончания блока кода, применяемой в других интерпретаторах. Естественно, многие существующие скрипты из-за этого в *Fish* не выполняются, но здравомыслие нового подхода компенсирует потери.

В наши дни большинство пользователей работает в оболочке через окно терминала, например, *Xterm* или *Konsole*, а не через лишённую графического интерфейса консоль. Поэтому приятно видеть, что разработчики постарались согласовать *Fish* с современными терминалами – данный интерпретатор отлично реагирует на изменение размера окна терминала, вмещая в него свой вывод таким образом, что не приходится спотыкаться об заломанные строки. К программе приложена документация в формате HTML, и по команде **fish --help** появляется окно *Konqueror* для её просмотра. Местами документация трудна для восприятия, но в ней заботливо предусмотрен раздел для новичков.

Мы будем счастливы, если *Fish* станет интерпретатором командной строки по умолчанию на дистрибутивах вроде Mandriva, ориентированных на новичков. И хотя *Bash* пустил слишком много корней, чтобы его легко было убрать с большинства систем, *Fish* предоставляет значительно лучший подход к освоению мощи командной строки, и разработчики явно приложили много сил, чтобы *Fish* реагировал на ваши действия так, как вы и ожидали. Этот интерпретатор должен знать каждый, даже те, кто годами пахал на **.bashrc** или являются самыми пылкими в мире поклонниками Zsh: он во многом поможет ускорить вашу работу. Отличный инструмент.



Как показано в правом окне, *Fish* отображает информацию о файлах на лету.

HOT GAMES

ТРЁХМЕРНАЯ КОСМИЧЕСКАЯ СТРЕЛЯЛКА

No Gravity

• **ВЕРСИЯ** 2.0 • **WEB** www.realtech-vr.com/nogravity

Звездная система OOLRO 3, 8002 год. Люди империи KROSSO, раскиданные по тысячам колоний, отпраздновали начало восьмого тысячелетия и теперь озабочены всплеском таинственных болезней. Некоторые говорят о надвигающейся эпидемии умственного заболевания… Пора полететь и перестрелять несколько вражеских космических станций.

Таков вкратце сюжет игры *No Gravity* (Невесомость), а коли это стрелялка, то игрок обязан решить зловещую проблему с помощью мощного оружия. Спасибо разработчикам, они снабдили свой продукт Автопакетом, поэтому не надо думать ни о компиляции, ни о дополнительных библиотеках: достаточно запустить файл **.package** для установки и ввести в терминале слово **nogravity**. Сразу бросается в глаза, что *No Gravity* добавила шику графике *Wing Commander'а* начала 90-х – но шрифт во всех меню и сообщениях, надо сказать, Абсолютное Зло (™). Неуклюжая попытка внести «крутизну» нечаянно привела к тому, что смотреть на игру физически больно.

И всё же, если вы удержите на игре воспаленный взгляд, то заметите, что *No Gravity* позаимствовала у *Wing Commander* не только графику. Расположенная в космосе зона игровых действий, где игрок может управлять несколькими космическими кораблями, подрывать вражеские корабли и выполнять миссии-задания – пусть это вторичное, но мы очень рады его возвращению. В игре пять эпизодов, каждый из которых содержит несколько заданий, так что скучать не придётся.

Некоторые взрывы представляют собой потрясающее зрелище; зря только разработчики засорили игру техно-музыкой. Звуковой фон сгодился бы для *Wipeout*, но во время космической битвы хочется слышать что-то более героическое, поднимающее боевой дух, например, Мусоргского, чтобы вдохновить вас на массовое избиение врага.

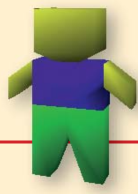


Типографская пародия в углу внизу справа – шрифт *No Gravity*. Читатели милосердно избавлены от ужасов саунд-трека.

No Gravity требует большей концентрации, чем львиная доля подобных стрелялок, зато система управления сделана на 5 баллов, и играть безумно весело. Со стилевыми огрехами надо бы разобраться, например, с омерзительными шрифтами и музыкой, немало убавляющими привлекательности, но попробовать игру непременно стоит.

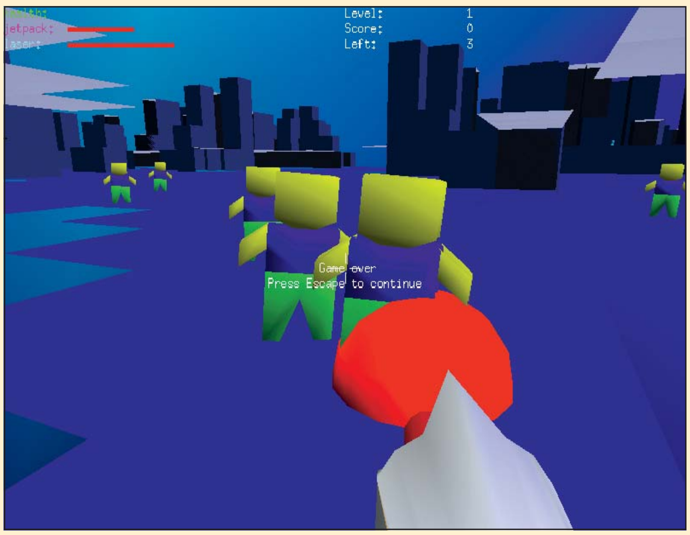
ШУТЕР ОТ ПЕРВОГО ЛИЦА

Enemy Lines 3



• **ВЕРСИЯ** 1.1 • **WEB** http://raum1.memebot.com/enemylines/part3.html

В конце 80-х бал правили платформенные игры. В начале 90-х на арену вышли стрелялки типа «убей-их-всех», которые потом уступили место гоночным симуляторам на 3D-приставках. Сегодня стрелялки от первого лица выглядят причудой, и поскольку разве что ленивая фирма еще не сбацала свой трёхмерный ускоритель, кажется, что жанр изжил себя. Рафаэль Пикрин (Raphael Pikrin), однако, спрятал пару карт в рукаве и наколдовал новаторскую *Enemy Lines 3* – третью в серии игр с открытым кодом, на сей раз названную «абстрактной трёхмерной стрелялкой».



Жизнь коротка в суровой стране из детского конструктора.

А что он подразумевает под словом «абстрактный»? Читай – «сюрреалистичный». На картинке слева вы не заметите ничего выдающегося: игра выглядит так, будто она писалась для Atari ST. Но *EL3* не сводится только к графике, по крайней мере не в ее традиционном понимании «набить экран текстурами». Речь идёт о странной атмосфере, холодной и мрачной, но непонятным образом создающей ощущение причастности.

Вас высаживают среди пейзажа с блочными строениями, домами и башнями, причем у некоторых прозрачные стены. Враги-роботы сыплются с неба и движутся по территории, угрожающе раскачиваясь по мере приближения к вам. С помощью лазера вы можете их уничтожить, прежде чем они дорвутся до контакта с вашим телом и снизят показатель вашего здоровья. Еще вам даден реактивный двигатель для перелетов через препятствия, и следите, чтоб у него не кончилась энергия.

На следующий уровень вы попадете, когда убьете всех роботов и соберете ключи, и всё это превосходно работает даже на старых компьютерах благодаря примитивности графики. В игре нет музыки, только звуковые эффекты, и, как ни странно, это помогает атмосфере игры – мелодия за кадром только разрушила бы чувство напряжённости и уязвимости.

EL3 работает, потому что она быстрая, неистовая, странная и нарезает круги как никакая другая. Жизнь в ней коротка, и играть следует скорее из любопытства, чем ради долгих приключений, но вас затягивает ее сумрачный мир. Страшно подумать о головоломных испытаниях, которые Пикрин мог бы сострапать, будь у него команда программистов вроде id…

FTP-СЕРВЕР

Bftpd

• **ВЕРСИЯ** 1.2 • **WEB** http://bftpd.sourceforge.net

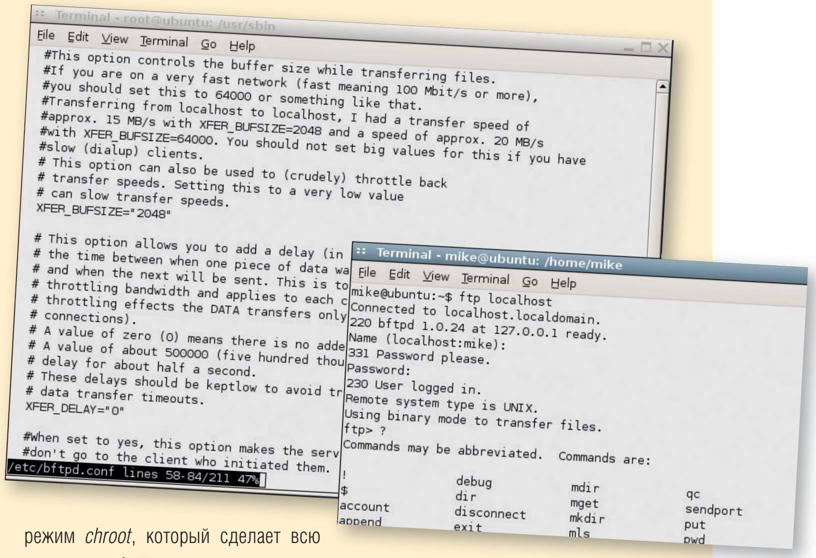
Скачивания через Web, сети P2P и неукротимый подъём *BitTorrent* оставили старую добрую систему FTP за бортом. Однако для неё всё ещё есть место в современном мире: она простая, быстрая и не требует кучи дополнительных программ для работы. Почти все операционные системы включают FTP-клиент, будь это просто команда *ftp* или графическое приложение, и новые FTP-серверы продолжают появляться. Так что встречайте *Bftpd* – это маленький, быстрый, простой в использовании и безопасный сервер, практически идеальный для повседневных задач.

Не имея дополнительных зависимостей, *Bftpd* может быть легко собран практически на любом дистрибутиве, при этом получится файл размером 50K. По умолчанию, программа попадет в **/usr/sbin**, с привязкой к файлу конфигурации **/etc/Bftpd.conf**. Желая упростить пользование *Bftpd*, разработчики проделали похвальную работу по облегчению чтения файла конфигурации. Каждая опция тщательным образом прокомментирована, и хотя кое-какое знакомство с FTP-сервером поможет делу, любой пользователь сможет его запустить и работать без особых хлопот.

> «BFTPD – ОТЛИЧНЫЙ ВЫБОР ДЛЯ ОБЩИХ ЗАДАЧ РАБОТЫ С ФАЙЛАМИ.»

Вы можете сконфигурировать порт, на котором работает FTP-сервер, назначить диапазоны портов для пассивного режима, а также установить режим *chroot*, который сделает всю остальную файловую систему недоступной в целях безопасности. Полезно, что предусмотрена настройка рабочих параметров, например, размер буфера и задержка передачи. Разработчики – молодцы, они дали системным администраторам много возможностей поиграть с настройками, а для обычных пользователей выбрали хорошие установки по умолчанию и максимально упростили процесс установки.

Сервер запускается в двух режимах: как самостоятельный демон, работающий на заднем плане в ожидании подключения пользователей, или через *(X)Inetd*, который запускает *Bftpd* при каждой попытке подключения. Работает *Bftpd* вполне предсказуемо: пользователи, имеющие учётную запись на хосте, могут входить в систему и переносить файлы на свои компьютеры командами *get* и *put*. Кстати предусмотрены и режимы пересылки ASCII- и двоичных файлов. В общем, *Bftpd* – не революционный сервер, но делает именно то, что должен. Отличный выбор для общих задач обмена файлами дома или в офисе.



Введите 'help' в *Bftpd*, и увидите список доступных команд.

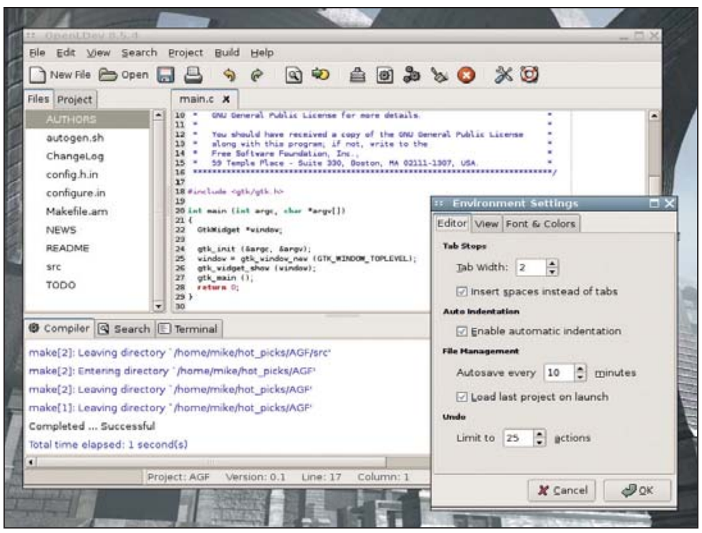
РЕДАКТОР ДЛЯ НАПИСАНИЯ ПРОГРАММ

OpenLDev

• **ВЕРСИЯ** 0.5.4 • **WEB** www.openldev.org

Эндрю Краузе, разработчик *OpenLDev*, считает, что большинство современных интегрированных сред разработки громоздки и сложны в использовании. Он задумывал *OpenLDev* как аккуратное и продуктивное ПО, задействующее инструментарий GNU, с акцентом на разработку программ на C и C++. Приложение построено на *GTK* и Gnome, поэтому вам потребуется версия *GTK*, начиная с 2.6, а также последние версии *libglade*, *libgnome*, *GtkSourceView* и *VTE*.

Раскладка основного окна *OpenLDev* используется большинством современных средств разработки: слева список для быстрого перехода между файлами, справа окно редактора, внизу окна панель с закладками для дополнительных функций, объединяющая вывод компилятора и встроенный терминал. Возможности редактирования довольно скудные, поддерживается лишь подсветка синтаксиса, нумерация строк, а больше почти ничего; нечего и сравнивать с редакторами, которые использует *Scintilla*.



Что впечатляет в *OpenLDev*, так это ведение проектов, толково согласованное со стилем разработки GNU. Например, можно создавать проекты на *C/C++* для командной строки или для *GTK*, а *OpenLDev* сгенерирует файл **main.c** и базовые файлы README и AUTHORS и воспользуется *Autotools* для создания сборочных скриптов. Все готово к написанию кода, без отвлечений на скучную бюрократию, и если вы решите распространять результаты своего труда, они будут выглядеть профессионально.

Для среды разработки, нацеленной на *GTK*, довольно странно, что при запуске программы *GTK* из *OpenLDev* всё со скрежетом зависает – правда, возвращается к жизни, когда вы закрываете своё приложение, но это изрядно раздражает, если хочется просмотреть код, пока программа работает. Да и средства отладки пригодились бы, а не только «собрал-запустил».

И все же для версии 0.5 *OpenLDev* проделывает солидную работу. Включены все необходимые функции и возможности, делающие программу хорошей легковесной интегрированной средой разработки. Чтобы основная программа не распухала, Краузе планирует интегрировать систему надстроек, и пользователи сами смогут добавлять такие функции, как поддержка *CVS* и *Glade*. Программистам здесь есть на что посмотреть.

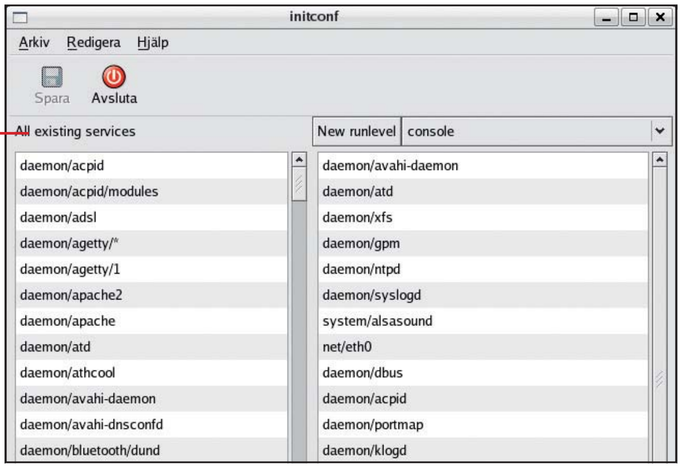
***OpenLDev* автоматически генерирует базовый код для написания GTK-приложений.**

HOTPICKS ПОВТОРНЫЙ ВИЗИТ

ЗАГРУЗКА СИСТЕМЫ

InitNG

- **ВЕРСИЯ** 0.5.3
- **WEB** http://initng.thinktux.net



***initconf* (пока на ранней стадии разработки) станет графическим интерфейсом Gnome для управления загрузочными скриптами.**

Не так уж и давно мы анализировали *InetNG* – в *LXF 72*, если быть точным – но с тех пор многое изменилось. *InitNG* досталась наша награда Hottest Pick за помощь в решении одной из самых одиозных проблем дистрибутивов Linux: медлительность процесса загрузки. Программа пока не включена ни в один из дистрибутивов, но многие пользователи уже подсадили ее в свои системы и теперь не нарадуются на полученное ускорение.

InitNG реструктурирует загрузочные скрипты Linux. В большинстве дистрибутивов скрипты запускаются последовательно, один за другим, ими так проще управлять, но отсюда и задержки. Например, ваша система ждет ответа от сервера DHCP, а могла бы в это время загружать модули. *InitNG* запускает скрипты параллельно, сокращая время загрузки более чем в два раза.

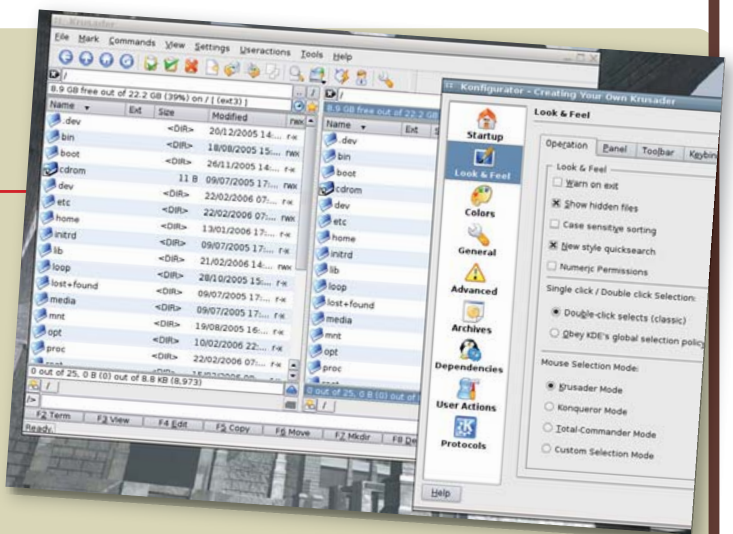
Со времени версии 0.1.6 список функций *InetNG* успел пополниться. Самые яркие из новинок – панель управления под Gnome, распознавание загрузочной системы дистрибутива *Slackware, Test_parser*, проверяющий корректность скриптов, и многое другое. Ещё более важно, что появилось множество новых «сервисов» – скриптов *InitNG*, посвященных решению специфических задач. То есть если вы пробовали *InitNG* ранее, но обнаружили, что некоторые вещи приходится запускать самостоятельно, например, соединение ADSL, на этот раз вам, возможно, повезет больше.

Мы следили за разработкой *InitNG* с самых ранних стадий, и нам приятно наблюдать прогресс от экспериментального набора скриптов до полной замены загрузочной системы Linux. Необходимы доделки, чтобы поддерживались все основные дистрибутивы, и мы надеемся, что версия 1.0 уже будет широко принята.

ФАЙЛОВЫЙ МЕНЕДЖЕР

Krusader

- **ВЕРСИЯ** 1.70.0
- **WEB** http://krusader.sourceforge.net



Панель конфигурации *Krusader*'а доводит процесс настройки до апогея – особенно по части скриптов и действий.

Много номеров вышло с тех пор, как мы делали обзор *Krusader* версии 0.79 в LXF 13 – несколько ледниковых периодов в развитии Linux! Этот файловый менеджер под KDE вышел вместе с KDE 1.x, когда по умолчанию был принят относительно примитивный *Kfm*, так чем же он сегодня ответит богатому функциями *Konqueror*? Разработчики *Krusader* не стояли на месте, и новый релиз 1.70 далек от своего предшественника.

Krusader предусматривает две панели для навигации по файловой системе, то есть вы может одновременно находиться в двух местах и копировать или перемещать файлы в любом направлении. На фоне раскладки по умолчанию *Konqueror* он выглядит перенасыщенным, но многим пользователям это нравится. Диапазон функций у *Krusader* поразительный: от синхронизации пользовательских профилей и каталогов до конфигурируемых действий пользователя и встроенного менеджера дисков.

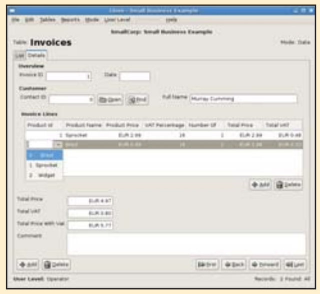
Вот некоторые из основных новых функций: «виртуальная файловая система», позволяющая отображать одновременно несколько результатов поиска; поддержка скриптов JavaScript с помощью *KJSEmbeded*; импортирование/экспортирование цветовой схемы; сравнение содержимого файлов; и многое другое. Это один из самых полных файловых менеджеров для Linux, и он все хорошеет.

KDE часто критикуют за встраивание чрезмерного количества опций, но по сравнению с *Krusader* и *Konqueror* выглядит консервативным. *Krusader* – не для всех, но если вы любите иметь под рукой большой объём операций над файлами, то не ошибётесь, выбрав его. Функциональность скриптов открывает его практически для любых задач. LXF

НОВЫЕ РЕЛИЗЫ

Новые и обновленные программы, заслуживающие внимания…

- ***XMMS2 0.2*** Переписанный плейер http://sourceforge.net/projects/xmms2
- ***DeVeDe 1.1*** Создание Video DVD www.rastersoft.com/programas/devede.html
- ***Sussen 0.15*** Сканер уязвимостей http://dev.mmgsecurity.com/projects/sussen/
- ***EZtorrent 1.2*** Публикация файлов в BitTorrent http://eztorrent.couriermta.com
- ***Glom 0.9.6*** Проектирование таблиц БД www.glom.org



- ***Prelude LML 0.9.3*** Анализатор лог-файлов www.prelude-ids.org
- ***Argo 0.2*** Монитор и контролер Xen www.steve.org.uk/Software/argo
- ***Bashish 2.0*** Темы для текстового терминала http://bashish.sourceforge.net
- ***DiskTool 1.6.0*** Консольный файловый менеджер www.mlsoft.org/dt.html
- ***Aide 0.11*** Система обнаружения вторжений www.cs.tut.fi/~rammer/aide.html
- ***Sparklet 0.9.1*** Многопользовательская



стрелялка http://sparklet.sourceforge.net

- ***BlogFS 1.0*** Монтирование WordPress через Fuse http://rohanpm.net/blogfs
- ***Asmutils 0.18*** Низкоуровневые утилиты http://asm.sourceforge.net/asmutils.html
- ***DB Browser 1.6.0*** Обозреватель структуры БД http://dbbrowser.berlios.de
- ***Aylet 0.5*** Плеер музыкальных файлов Spectrum http://rus.members.beeb.net/aylet.html
- ***Packer 0.1.5*** Создатель RPM/.deb/Autopackage http://packer.sourceforge.net

Что такое... SOLID?



Неудобное управление устройствами является слабостью Linux-систем. Программисты KDE придумали изящное решение этой проблемы. **Джоно Бэкон (Jono Bacon)**, засучив рукава, исследует Solid.

>> Итак, что это – Solid?

Solid – значимая составная часть будущего KDE 4.0. Проект KDE 4.0 был разделен на несколько различных компонентов: Solid, Plasma, Oxygen и Phonon, каждый из которых выполняет свою работу на рабочем столе.

>> И что же Solid делает?

Solid отвечает за взаимодействие с оборудованием. Традиционно, легкое управление устройствами в Linux было равносильно просьбе одолжить золотой медальон у мистера Т: это возможно, но невероятно, и, скорее всего, вызовет неприятности. Solid направлен на устранение этого недостатка.

>> Как работает Solid?

Основная цель Solid – предоставить адекватный программный интерфейс для оборудования. Во времена KDE 1.x было невероятно сложно взаимодействовать с оборудованием в приложениях. Общение с устройствами осуществлялось индивидуально в каждом приложении. Например, если вы писали приложение для обмена информацией с цифровыми камерами, то вам пришлось бы включать в него код, ответственный за общение с камерой.

>> Звучит как-то дико. А если другим приложениям тоже нужно общаться с камерой?

Вот именно. В этом и была проблема поддержки устройств в приложениях – вместо сосредоточения на логике приложения, разработчикам приходилось заниматься спецификой оборудования. Различия в дистрибутивах, разные версии ядра Linux и другие факторы подбавляли сложностей. Кроме того, существовала проблема горячего подключения устройств. Вставив USB-устройство в компьютер, вы ожидаете появления иконки на рабочем столе. В Linux это делается совершенно не так, как в Solaris, BSD, Mac OS X и Windows.

>> Надеюсь, Solid решит все эти проблемы и избавит нас от лишних хлопот?

По большей части да. Задача Solid состоит в том, чтобы предоставить унифицированный программный интерфейс (API) к оборудованию. Вместо того, чтобы включать в приложение специфический для оборудования код, Solid позволит абстрагировать оборудование. Таким образом, другой инструмент будет выполнять работу по обмену информацией с устройствами.

>> Что значит «другой инструмент»?

Что-то наподобие HAL, Hardware Abstraction Layer [прим. перев.: HAL – уровень аппаратных абстракций].

> «HAL – СРЕДСТВО ОБМЕНА ИНФОРМАЦИЕЙ С ОБОРУДОВАНИЕМ, А SOLID – ИНСТРУМЕНТ ОБРАБОТКИ СОБЫТИЙ НА РАБОЧЕМ СТОЛЕ.»

HAL – это демон, который отслеживает аппаратные изменения в конфигурации системы и представляет эту информацию в форме, понятной приложениям. По существу, инструмент HAL дает вам ясное программное представление, как выглядит ваше оборудование. HAL, в свою очередь, полагается на D-BUS, который обеспечивает взаимодействие разных подсистем приложения. Например, если вы хотите синхронизировать свою музыкальную коллекцию с плейером, то можете использовать HAL и скрипт на Python.

>> Я не совсем понял, зачем тогда Solid? Почему нельзя просто обойтись HAL?

На это есть пара причин. Первая: проект KDE всегда старался поддерживать разные операционные системы, не ограничиваясь только Linux. По существу, должна быть поддержка Linux, BSD, Mac OS X, Windows, Solaris и других. Так как HAL полагается на специфические особенности ядра Linux, то он используется только в Linux. Ядро предоставляет информацию об оборудовании в Sysfs, которая в свою очередь используется *udev*, далее она используется D-BUS, которая затем используется HAL.

>> То есть Solid замещает HAL кросс-платформным эквивалентом? А так бывает?

Нет, это не входит в намерения разработчиков KDE. Не полагаясь исключительно на HAL, Solid будет поддерживать множество различных «основ» (back-end), которые зависят от операционной системы. На платформе Linux, Solid использует HAL для обмена информацией с устройствами, но на Solaris, BSD, Mac OS X и Windows, Solid будет использовать аналоги HAL.

>> И такие аналоги есть?

На одних системах есть, на других – нет. Если вы хотите узнать об аналоге HAL в BSD, возможно, стоит почитать *BSD Format*. Хотя, погодите...

>> Вы сказали, есть пара причин, почему HAL нельзя просто заменить Solid. В чем вторая?

У Solid и HAL немного разная идеология. Главная задача HAL – предоставить информацию о доступных устройствах в системе. Если вы хотите использовать эту информацию, чтобы как-то повлиять на рабочий стол, то придется самим писать код приложения. Собственно, даже и с чем-то вроде HAL вам придется писать много лишнего кода для обработки события при подключении устройства.

>> Приведите пример...

Предположим, вы подключаете USB-брелок и хотите, чтобы появилось окно, отображающее список файлов

на этом устройстве. Тогда вам понадобиться написать код, который вызывает окно в ответ на событие, возникающее при подключении устройства, которое генерирует HAL. Другой пример – цифровые камеры. Когда вы подключаете камеру к системе, то хотите воспользоваться функциональностью, характерной для цифровой камеры и KDE. В этом случае пригодится Solid.

>> То есть Solid – это API KDE, отрабатывающий сценарии для оборудования?

Верно. Можно считать, что HAL – средство обмена информацией с оборудованием, а Solid – инструмент для обработки событий на рабочем столе. По существу, Solid может не только работать с аналогами HAL, но и предоставляет независимый от операционный системы API, позволяющий обрабатывать различные события, связанные с устройствами, в приложениях KDE.

>> Хорошо. Как все это работает технически?

KDE базируется на наборе инструментов *Qt* от Trolltech. Qt базируется на системе сигналов и слотов: когда в каком-то приложении что-то происходит, оно может запустить сигнал. Например, при нажатии кнопки инициируется сигнал **clicked()**, который можно присоединить к слоту. Слот – определенная программистом функция, которая будет обрабатывать сигнал. Solid предоставит множество сигналов, которые будут говорить об изменениях в оборудовании, а разработчики смогут просто написать слоты для их обработки.

>> Зачем Solid поддерживать Windows и Mac OS X? Вряд ли им нужен KDE...

Qt доступна на Linux и Mac OS X по лицензии GPL уже многие годы, а Windows-версия Qt выпущена под GPL в феврале 2005 года. Это событие открыло дорогу приложениям KDE на другие операционные системы и дало разработчикам новый импульс. Команда разработчиков KDE намерена создать самую лучшую рабочую среду, и хотя многие из разработчиков, конечно, пользуются Linux, было принято соглашение о создании платформы, которая сможет работать на максимальном количестве ОС. Теперь, когда Qt стала свободной на всех основных платформах, выполнять соглашение будет проще.

>> Пока все это слова, а в каком состоянии находится сейчас эта разработка?

В настоящий момент напряженная разработка Solid ведется под руководством Кевина Оттенса (Kevin Ottens); написан код для некоторых классов устройств (например, CD-ROM), а также заготовлено некоторое количество сигналов. Несмотря на текущее положение дел, Solid находится в состоянии разработки и уже вписан как ключевая особенность KDE 4. Выпуск KDE 4 в ближайшее время не ожидается, поэтому пройдет некоторое время, прежде чем вы увидите Solid на рабочем столе.

>> Где об этом узнать поподробнее?

Большая часть работы по Solid ведется через главный сайт – **http://solid.kde.org**. Дополнительно можно посмотреть рассылки разработчиков в **lists.kde.org**. Главная дискуссия, где обсуждают Solid, происходит в Kde-hardware-devel.

Если вы мыслите настоящим временем, информацию можно почерпнуть и на IRC-канале #solid сети Freenode IRC. Наконец, больше всего о разработке Solid и других технологий, относящихся к KDE 4, можно узнать на Planet KDE (**www.planetkde.org**). Этот сайт содержит заметки разработчиков KDE и позволяет понять текущее состояние проекта. Там вы можете узнать, какие технологии приняты на вооружение, а от каких отказались. LXF





TOP TRUMPS COURTESY: WINNING MOVES UK LTD

Визитка LXF

Брюс Пиренс

Основатель группы Software in the Public Interest, основатель Open Source Initiative, основатель Linux Standard Base и основатель UserLinux – Брюс внес значительный вклад в процесс основания чего-либо.

ВОЗРАСТ	48
НАЦИОНАЛЬНОСТЬ	Американец
ОПЫТ РАБОТЫ В UNIX	13
ЯЗЫКИ ПРОГРАММИРОВАНИЯ	30
КОЛИЧЕСТВО ПК	13 (8 ноутбуков)
ДНЕВНАЯ НОРМА КОФЕ	1 чашка
ПАР САНДАЛИЙ	1

ПРЯМАЯ РЕЧЬ «Если на вашем VCR все еще мигает 12:00, то Linux вам ни к

БРЮС впередсмотрящий

Брюс Пиренс (Bruce Perens) не такой большой, как на фото, но тень он отбрасывает достаточно длинную. Мы побеседовали с бывшим лидером проекта Debian о свободе, Ubuntu и… *кхе-кхе*… Red Hat.

LXF ИНТЕРВЬЮ

Некоторые люди сидят по ту сторону стены, где программы распространяется бесплатно, некоторые – по другую, где открыт исходный код. Но лишь немногим – таким, как Брюс Пиренс – удалось побывать по обе стороны, и они понимают, насколько лучше стало бы всем нам, если б стены не существовало вовсе, и стараются соединить обе философии. Вы могли встречать Пиренса в самых различных местах: одно время он возглавлял проект Debian, он основал Open Source Initiative, и под его именем даже вышло в свет несколько книг по проблемам технологии. Мы встретились с Брюсом в Сан-Франциско и расспросили его о том, какой дистрибутив Linux он предпочитает, о путанице с лицензиями и о будущем UserLinux…



Linux Format: **Чтоб представить Вас нашим читателям, расскажите, пожалуйста, вкратце о себе.**

Брюс Пиренс (БП): Наверное, прежде всего я известен как создатель определения термина Open Source. Это – манифест Open Source, устанавливающий правила его лицензирования. И этому манифесту следуют до сих пор. В этой работе я стою на плечах гигантов – Ричард Столлмен (Richard Stallman) самый крупный из них. Ричард проделал колоссальную работу, которая привела нас к настоящему положению дел и которую просто невозможно переоценить.

Еще я выпустил серию книг – на сегодняшний день их 17, все они изданы в Prentice Hall. Все книги вышли по открытой лицензии – вы можете купить их, накопировать и торговать копиями. Пока копий никто не продавал. Но я создал незашифрованные PDF-документы для скачивания. Некоторые ими пользуются, книг не покупают, но я не думаю, что от этого наши продажи серьезно пострадали. Ну любят люди бумагу.

Я преподаю в двух школах, я профессор в колледже Университета Адгер, Кристиансен, Норвегия (Agder University College), и старший научный сотрудник Университета Джорджа Вашингтона (George Washington University) в Вашингтоне, округ Колумбия. Приблизительно раз в год я бываю на GWU, а в Норвегии значительно чаще.

LXF: **И где Ваша база?**

БП: Беркли, Калифорния. И еще я вице-президент Sourcelabs, это новое предприятие на венчурном капитале, призванное оказывать поддержку ПО с открытым кодом на предприятиях. На данный момент самый важный их продукт – пакет на Java, содержащий большинство компонентов с открытым кодом, замкнутых на крупные предприятия, штуки вроде Hibernate и т.д.

LXF: **А еще Вы побывали ГПД [Главой проекта Debian] и написали Free Software Guidelines для него.**

БП: Ну, на самом деле Free Software Guidelines для Debian превратились в Определение Open Source. Но в Debian я пришел, когда там было примерно 60 человек, а Иэн [Мердок] (Ian Murdock) еще учился в колледже и вел этот проект. Я чувствовал, что Debian – очень важная работа. Главной моей мотивацией в тот момент являлось желание создать систему Linux для радиолюбителей, соединив два нуднейших занятия – в итоге это сделали без меня, есть версия Knoppix для радиолюбителей.

Но так уж вышло, что Debian я занимался больше, чем радио, и превратился в руководителя проекта. Я и сейчас участвую в проекте, так что, наверно, я участвую в нем практически всю его жизнь, и сейчас я вхожу в правление Debian.

LXF: **Похоже, от Debian пошло немало ответвлений – многие взяли Debian за основу.**

БП: Debian GNU/Linux по определению обязан иметь производные.

LXF: **А Вы, похоже, увязли в самом Debian – не перешли ни на Red Hat, ни на Morphix, ни на Knoppix…**

БП: Вы знаете, я однажды плюнул на парней из Debian и попробовал использовать в своих системах Red Hat, но очень скоро опять вернулся к Debian.

LXF: **И давно Вы пробовали?**

БП: У меня на ноутбуке стоит Debian. Давно ли я пробовал Red Hat? Ну, как-то в одной из должностей я посидел в системе Red Hat, я не думаю, что с Red Hat что-то не так, просто ее нельзя сравнить с Debian ни по широте предлагаемых приложений, ни по количеству разработчиков.

LXF: **Ваше участие в Debian было очень активным – Вы говорите, что Вы до сих пор часть этого проекта, но вряд ли Вы так же активны, как раньше…**

БП: Да, я уже не милостивый диктатор...

LXF: **Насколько велики перемены?**

БП: Ну, я бы сказал, огромны. Меня зауважали за то, что я увеличил команду Debian с 60 человек до 200, а сегодня в ней больше 1000 человек. Буквально на всех континентах сидят разработчики Debian, который, по моему мнению, был и останется важной частью Linux [как явления, – прим.ред.]. И я полагаю, что со временем он станет более важен в коммерческом отношении – мы уже видели несколько неудачных попыток вывести его на коммерческий уровень, включая попытки, которые предпринимал я сам. Я нанял Иэна, чтобы он работал с Progeny, которая изначально задумывалась как фирма Debian, но это не сработало, рынок был еще не готов. UserLinux может все-таки добиться успеха, а Debian Common Core Consortium воплощает множество принципов UserLinux. Получится ли что-нибудь из проекта UserLinux или нет – все равно многое из того, что я задумал для этого проекта, будет существовать и дальше.

LXF: **Проект Debian, кажется, сильно замедлился в своем развитии; этот Sarge…**

БП: Подготовка релиза Sarge заняла у Debian ужасно много времени, а одной из причин стало изменение системы установки. Это была действительно серьезная работа, очень серьезная, и на предыдущих релизах этой работы старательно избегали, но наконец решили: «Все, пора браться за дело, пока не решим эту проблему, ничего не получится».

Сейчас Debian сконцентрировал свое внимание на 18-месячных релизах, но мне бы хотелось сократить это время до одного года, хотя, наверно, я хочу слишком много. Я думаю, все работающие в проекте Debian отдают себе отчет, что если через 18 месяцев релиз не выйдет, их никто не будет воспринимать всерьез.

О СРОКАХ РЕЛИЗА

> «Все в проекте Debian знают, что если через 18 месяцев релиз не выйдет, их никто не будет воспринимать всерьез.»

LXF: **Когда Вы думаете об Open Source Initiative как о концепции, Вам видится множество открытых лицензий?**

БП: Нет. Если бы я воображал такое множество открытых лицензий, я бы формулировал правила по-другому.

Слишком богатый выбор, от которого глаза разбегаются, – практически всегда плохая идея. Я думаю, когда мы только начинали, мы потому хотели получить лицензии от множества самых разных крупных фирм, что это позволило бы нам говорить: «Вот у нас лицензия IBM, а вот – лицензия Sun», и репутация этих фирм распространялась бы на нас. Сейчас ясно, с точки зрения полезности для сообщества это была очень, очень плохая идея, и я так понял, что Open Source initiative работает над проблемой расплодившихся лицензий. У них есть для этого комитет, но меня туда не приняли – я подавал заявление на участие в его работе, а в ответ получил почти стандартный бланк, где говорилось: «Извините, Ваша кандидатура не прошла».

LXF: **Наверно, остались отрицательные эмоции.**

БП: Ну, я думаю, что в совете директоров OSI есть парочка человек, которые… скажем, недостаточно повзрослели.

LXF: **Поговорим о UserLinux. Вы объявили о нем, так? Год, два года назад?**

БП: Мы работаем над ним почти два года. Web-сайт по-прежнему существует, но проект смахивает на город-призрак. Мне кажется, его можно реанимировать; сейчас, когда я работаю с Sourcelabs, я могу снять с себя кое-какую нагрузку и посвятить побольше времени UserLinux.

Для меня проблема с UserLinux заключалась в том, что когда я начал работу над этим проектом, я был независимым предпринимателем, то есть консалтинговой фирмой в одном лице. Так что приходилось крутиться и заниматься коммерцией, чтобы зарабатывать на жизнь каждый месяц, одновременно будучи лидером движения Open Source и работая над большим количеством проектов. Времени на все просто не хватало, и в какой-то момент…мне никто не платил за работу над UserLinux, а первым делом надо было обеспечивать семью.

А теперь у меня фирма, которая платит мне за работу, которую я считаю нужной для движения Open Source, на условиях частичной занятости, так что я могу теперь уделять этой работе больше времени. Кое-что я скидываю с плеч долой: я только что создал модель безопасности для Ruby On Rails, и вместе с главными разработчиками Ruby On Rails работаю над созданием релиза, с которым смогут работать нормальные люди – пока что для этого надо быть гуру. Еще я должен расквитаться с одной книжной работой, и это – два основных проекта, покончив с которыми, я смогу больше времени уделять UserLinux.

LXF: **После запуска UserLinux Ubuntu, похоже, взялось играть ту же роль. Я думаю, Джефф Во (Jeff Waugh) сде-**

>>



«Итак, Брюс, чем отличается свободное ПО от ПО с открытым исходным кодом?»

лал хороший ход, сказав: «Эй, парни из UserLinux, можете работать с нами над Ubuntu».

БП: Я и в самом деле собирался поработать с Canonical, когда Марк [Шаттлворт] (Mark [Shuttleworth]) запускал эту работу и у них была пара проблем. Я думаю, Марку интересно, чтобы у него была успешная и прибыльная фирма, и мне не кажется, что дистрибутивы Linux подходят для коммерческой фирмы. На моем сайте есть длинная статья про экономику открытого кода [http://perens.com/Articles/Economic.html], и из этой статьи вы можете почерпнуть факт, что открытый код наносит наибольший ущерб коммерческим дистрибутивам Linux.

Так что я чувствую, что в дальней перспективе большая часть работы по поддержке дистрибутивов Linux вернется в ту сферу, где командует пользователь, где пользователи, заинтересованные в добавлении какой-то функциональности, нанимают исполнителей, платят им за работу, ну и т.д. Debian всегда следовал этой модели, и мы уже наблюдали несколько вялых попыток ввести то хорошее, что есть в Debian, в другие дистрибутивы; Fedora и Novell собираются сделать нечто подобное. А проблема у всех этих попыток одна – управление. Знаете, Майкл Тиманн (Michael Tiemann) пытался сделать Fedora независимой, но не будет она такой же независимой, как Debian, никогда.

А насчет Canonical, что меня поразило в Марке – это что он желает иметь тотальный контроль. Например, когда я собирался устроиться к нему в фирму, одним из препятствий стал тот факт, что Марк не дает своим сотрудникам акций в своих фирмах. Если я работаю на кого-то, я хочу быть в какой-то степени участником этого дела, так что я полагаю, что хотя Ubuntu и Canonical могли бы многое сделать для Debian и стать отличными членами сообщества, они никогда не станут его ядром, а мы могли бы стать ближе к ядру, следуя тем правилам, что я сформулировал для UserLinux.

LXF: **Как, по вашему мнению, сказанное выше относится к Red Hat и Novell на дальней перспективе? Модель Red Hat – создать хороший дистрибутив и зарабатывать на поддержке.**

БП: [вздыхает] Я общаюсь со многими клиентами, потому что у Sourcelabs есть клиенты. Не считайте это официальным заявлением Sourcelabs, у Sourcelabs нет проблем с Red Hat: но каждый клиент, с которым я общался, Red Hat ненавидит. Я думаю, как раз из-за сервиса.

LXF: **Вы считаете, что их бизнес-модель под угрозой?**

БП: Я считаю, что коммерческая фирма – неправильная форма для создания дистрибутивов Linux. Потенциально правильно заниматься обслуживанием, но только в том случае, когда имеется рынок обслуживания, а не монополия на обслуживание.

LXF: **Поговорим о Sourcelabs. Вы уже упоминали эту фирму: это она оплачивает половину Вашего времени, чтобы Вы занимались тем, чем хочется...**

БП: Благодаря им я отдал своего сына в частный детский сад, это одна из основных причин, почему я пошел на эту работу. Я работаю над тем, что люблю, 100% моего времени, а половину этого времени сам и планирую: захочу, буду программировать для Ruby On Rails, захочу – отправлюсь в Европу, чтобы выразить протест против патентов на программное обеспечение, для меня это одинаково возможно.

Со второй половиной своего времени я на данный момент в большей степени определился: Sourcelabs – молодая фирма, и я помогаю им сотрудничать с фирмами списка Fortune 100. Мы не обнародуем список клиентов, но я могу сказать, что это – целый город мирового уровня, это большое количество крупных фирм, чьи названия известны всем.

LXF: **Дело в том, что, по-моему, Sourcelabs – очередная фирма- новодел из пытающихся вернуть бум доткомов. Огромный стартовый капитал, много великих идей, тут и открытый исходный код, а чем фирмы занимаются – неясно.**

БП: Неразберихи хватает, и хватает фирм, которые декларируют одно и то же. А отличить от них Sourcelabs вы можете – и это было очень важно для меня – потому, что Sourcelabs не закрывается ни от кого. Когда Sourcelabs выпускает новую программу, эту программу можно найти на их сайте. Сейчас они спрашивают ваше имя и адрес, чтобы присылать вам рекламу, но зато вы получаете программу, и, естественно, поскольку эта программа распространяется по открытой лицензии, вы можете сделать зеркало этого сайта и не спрашивать ни у кого ни имени, ни адреса.



Брюс в президиуме на OSCon 2005, рядом (слева направо) Бернард Голден (Bernard Golden), Робин Миллер (Robin Miller) и Джон Терпстра (John Terpstra).

Считается, что открытое ПО проходит большое количество тестов, но сейчас тестирование в Open Source преимущественно модульное, за последние несколько лет мы это хорошо отработали, зато не тестировали масштабируемость, не тестировали систему в целом и т.д. – а без этого мы не вправе заявить: «Стройте на ней свой бизнес, она вас не подведет».

Так что Sourcelabs в основном создала рынок поддержки, и попыталась стать первым гражданином на этом рынке. Никто не говорит, что другая фирма не может заниматься поддержкой того же самого ПО. Мы считаем, что это как раз в интересах наших клиентов, и мы надеемся, что клиенты выберут нас по этой причине, а еще по той причине, что у нас много отличных специалистов.

На данный момент в фирме только трое сотрудников не являются инженерами, и когда вы обратитесь за поддержкой в Sourcelabs, с вами будет общаться не читатель скриптов из Индии, а программист, который работает именно с тем ПО, с которым у вас проблемы, и он вполне может оказаться одним из ключевых разработчиков. По мере расширения фирмы мы рассчитываем работать таким образом со все большим количеством проектов, и мы думаем о будущем. Например, я разработал экспертизу по Ruby On Rails, и мы полагаем, что это – очень многообещающая платформа. Пока что мы не поддерживаем ее официально, в той же области много другой работы для экспертизы, и фирма может продвигаться на рынке.

***LXF:* Итак, какой-нибудь SpikeSource наготовит открытый код…**

БП: Мы не знаем, каков бизнес-план SpikeSource; и думаем, что они и сами толком этого не знают. Мне нравится Ким [Полезе] (Kim Polese), она – приятный человек. У них значительный капитал, они трудоустроили много народа, а в мире Open Source не принято поливать конкурента, который приносит пользу сообществу, не такие мы люди. Поэтому я надеюсь, что SpikeSource и Sourcelabs будут вроде HP и IBM: и те, и другие приносят пользу. Пока я не могу ничего сказать о том, чем они занимаются, но в скором будущем это станет ясно.

***LXF:* А как Вы смотрите на организации, помогающие сообществу Open Source – например, OSDL?**

БП: Да, некоторые фирмы пытаются помочь сообществу Open Source. Но у OSDL на это есть серьезные препятствия, потому что интересы многих членов этой организации противоречат интересам сообщества. Большинство этих ребят принадлежат к лагерю, выступающему за патенты на ПО. Эти патенты – оружие номер один против сообщества Open Source, и мы должны бороться с ними сообща. К сожалению, OSDL тут помочь не в силах. Я обсуждал эту проблему со Стью [Коэном – Stu Cohen, исполнительный директор OSDL], и думаю, что он отлично понимает мою точку зрения.

Я очень рад, что OSDL вложилась в предприятие Эбена Моглена (Eben Moglen), это нечто вроде юридической помощи сообществу открытого кода, где вы можете получить компетентную консультацию, чего нет у Slashdot! Я думаю, достойно восхищения, что компании начинают слегка вытягивать шею в сторону политики. Например, мы видим, что MySQL оказывает финансовую поддержку NoSoftwarePatents.org, это очень успешная кампания в Европе, благодаря которой конфликт на несколько лет притих – хотя и нельзя говорить о нашей полной победе в нем. И очень хорошо, что Sourcelabs дает мне возможность говорит подобные вещи, и при этом не указывает мне, что я должен сказать.

***LXF:* И Вы сможете бороться с патентной реформой в США?**

БП: Я думаю, сейчас это становится возможным; думаю, мы подходим к той точке, когда представители фирм начинают понимать, что система уже дала трещину. В этом году мы получили Закон о Патентах [2005], который не очень-то помогает решить проблему – в основном он на стороне крупных патентовладельцев.

***LXF:* Болтали, что ключевые фигуры, вроде Вас или Ричарда Столлмена, уже не нужны и сообщество открытого кода само себя продвигает. А Вы что об этом думаете?**

БП: Я всегда считал себя не лидером, а человеком, который подает голос в защиту интересов сообщества, причем на основании писем, которые я получаю. Когда я публикую статью или общаюсь с прессой, мне часто пишут и говорят: «Спасибо, что Вы сказали об этом. Я чувствовал то же самое, но не умел объяснить окружающим. А теперь я могу показать им вашу статью». Забавно, но когда я учился, я выбрал специализацию по искусству общения, а не в инженерной сфере, и поэтому, видимо, тут я сильнее.

Я думаю, что сейчас люди вроде меня или Ричарда Столлмена нужнее, чем раньше, потому что если бы нас не было, вакуум заполнили бы пресс-атташе разных корпораций, и они выступали бы в защиту собственных интересов, а не интересов сообщества Open Source. Уверяю вас, что мы не такие. LXF

О RED HAT

«Все, с кем я общался, ненавидят Red Hat. Я думаю, это из-за обслуживания.»



Другие мысли Брюса – о поддержке AMD64 для Debian, об отцовстве, о проблемах с OSI – читайте на www.linuxformat.co.uk/mag/perens.html.

ВСЕ В MEPIS!

Меняете свою ОС? Сделать правильный выбор трудно, но пока вы не приняли решение, почему бы не провести выходные с MEPIS? Ваши гиды: Ник Вейч, **Пол Хадсон** и **Майк Сондерс**.

НА ДИСКЕ

Хотите ли вы использовать Linux в дороге на ноутбуке или дома на рабочем столе, у SimplyMepis есть что вам предложить. Вобрав в себя простоту использования Mandrake, стабильность Debian и гибкость Knoppix, SimplyMepis стремительно набирает популярность. Если считать, что ночь четверга – это новая пятница, а серый – новый черный, то SimplyMepis – это новый Ubuntu.

Если вы используете Linux несколько лет, то, вероятно, намного легче держаться привычного дистрибутива, даже если вам не все в нем нравится. Но стоит пробовать и что-то новое. SimplyMepis обладает множеством привлекательных особенностей – для начала, он включает целую тучу проприетарных библиотек и расширений.

Почему это так здорово? Ну, в прошлом вы могли огорчаться, попытавшись запустить игру и обнаружив, что графический драйвер работает некорректно. Или – вы посещали web-сайты, из-за использования Flash казавшиеся пустыми; или тратили часы на загрузку AVI-видео, а в итоге обнаруживали, что посмотреть его нельзя. SimplyMepis пытается исправить положение, предварительно установив и настроив богатый набор проприетарных модулей, кодеков и драйверов. Поэтому почти все должно заработать (за исключением зашифрованных DVD).

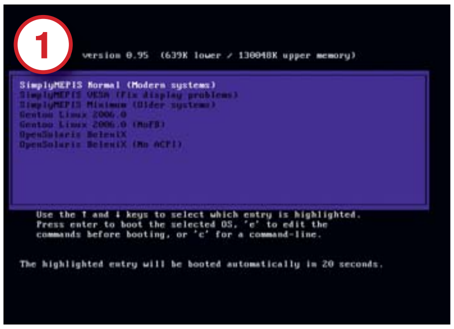
Рассмотрим использование сети. SimplyMepis просканирует доступные устройства, попытается настроить DHCP и использует его для получения всех необходимых параметров. Поскольку это охватывает не меньше половины всех компьютеров, которые, возможно, используют Linux, это хороший первый шаг.

Но действительно прекрасно, что вам не надо кому-то верить на слово. SimplyMepis поставляется как LiveCD, так что просто вставьте диск в ПК – и все. Если вы решите установить его на жесткий диск, можете запустить установщик прямо изнутри SimplyMepis. Следующие несколько страниц описывают способы запуска и установки SimplyMepis, а также причины, благодаря которым у вас может возникнуть желание им пользоваться, и заканчиваются интервью с создателем дистрибутива Уорреном Вудфордом (Warren Woodford).

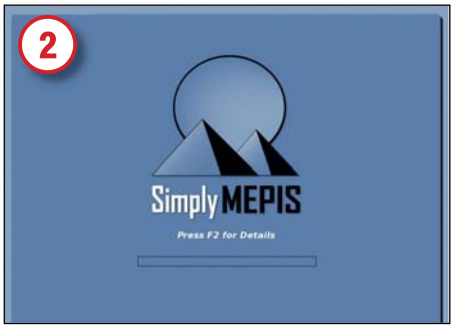
Начнем

На большинстве современных ПК запуск SimplyMepis настолько прост, что проще не бывает: просто вставляете DVD-приложение в ваш привод и перезагружаетесь. Так как это Live-дистрибутив, то система грузится с DVD/CD, не изменяя содержимое жесткого диска, и если что-то пойдет не так, можете просто перегрузить машину, вынуть диск и вернуться к прежней ОС. Для хорошей производительности вам необходим процессор с частотой 500 МГц и 256 МБ или более оперативной памяти. Далее следует краткое введение...

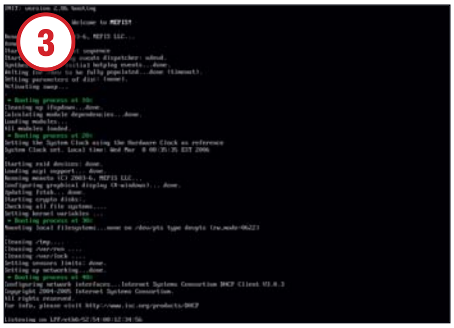
КАК ЗАПУСТИТЬ SIMPLYMEPIS С LIVECD



1 Перегрузив ПК при наличии в приводе DVD-приложения к журналу, вы увидите это меню. (Если это не так, нужно настроить вашу машину на загрузку с DVD/CD вместо жесткого диска). Обычно при появлении выделена система SimplyMepis, так что просто нажимаем клавишу Enter, и процесс пошел.



2 Через несколько секунд появится эта заставка, а SimplyMepis будет загружаться. Полоска-индикатор под логотипом показывает, сколько осталось до завершения загрузки; на современных ПК потребуется от одной до двух минут. Если индикатор остается неизменным более двух минут, нажмите F2 для получения полной информации о загрузке...

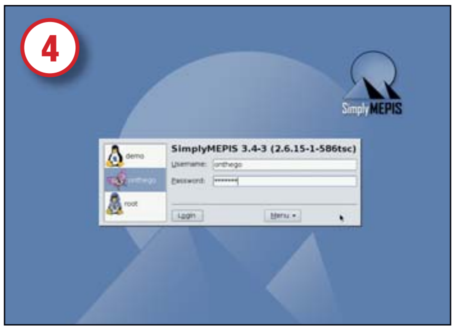


3 ...как показано здесь. А показано здесь все, что происходит в процессе загрузки, поэтому если произошла остановка или возникает повторяющееся сообщение об ошибке, запишите его и попросите помощи у ветеранов Linux (например, в форуме *LXF*). Если выдается сообщение об ошибке *X* GUI, перезагрузите ПК, вернитесь к шагу 1 и выберите режим SimplyMepis VESA.



НОВИЧОК В LINUX?

Установка Linux может казаться новичкам непростым делом – люди бормочут о дистрибутивах, пакетах, *Grub* и о всяких штуках, про которые вы и не слыхивали. Но в SimplyMepis вы можете попробовать приложения прямо с CD – за это он и называется LiveCD. Другими словами, можно запустить операционную систему без форматирования жесткого диска или риска потерять важные файлы. SimplyMepis легко справляется с автоматическим определением и настройкой оборудования, так что он без проблем заработает на любом достаточно современном компьютере. Просто вставьте DVD-приложение с SimplyMepis в привод и дайте ему раскрутиться. Если что-то тормозит, то исключительно из-за обращений к диску.



После загрузки компьютера вы плавно въедете в окно входа в систему. Уже зарегистрировано три пользователя – щелкните слева на иконке Demo и введите пароль 'demo', после чего нажмите Enter для входа. Пока вам это не нужно, но на будущее запомните пароль пользователя onthego – 'onthego' (см. стр. 48), а пароль для root (администратора) – 'root'.



Можно начинать исследование SimplyMepis Linux и его программ! Помните, что система, запущенная с DVD/CD, работает не так быстро, как с жесткого диска, но это можно изменить, следуя инструкциям на стр. 49.



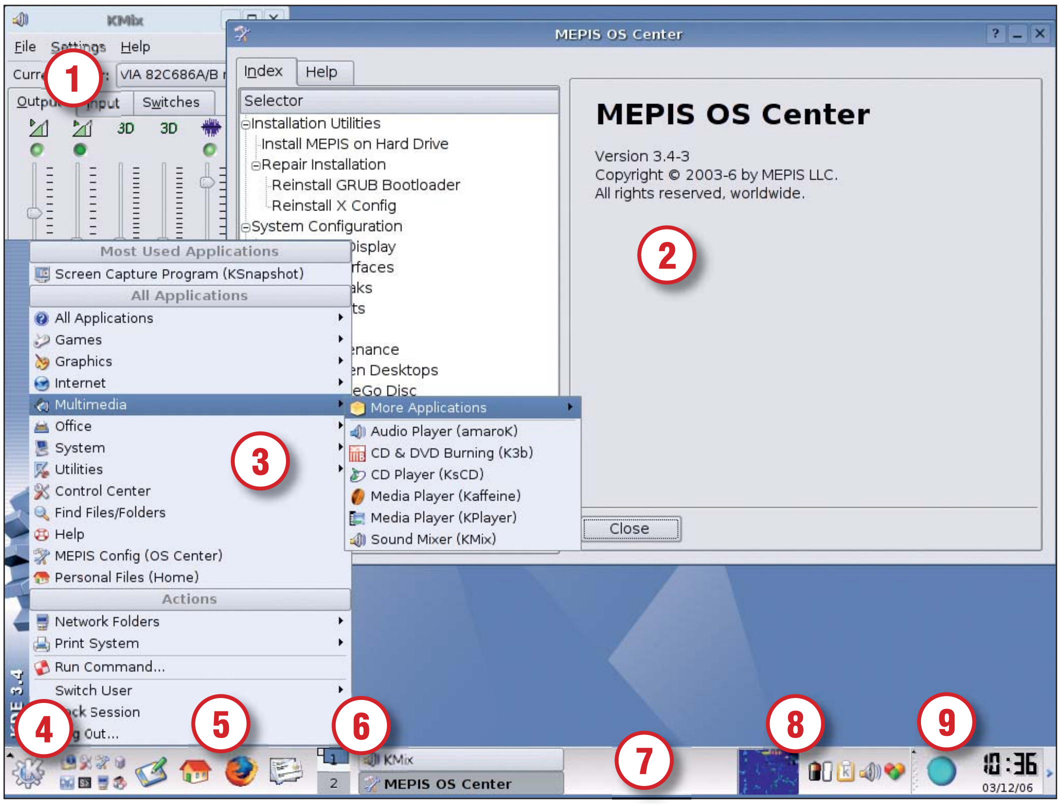
Если хотите вернуться к своей обычной ОС, щелкните на кнопке K-меню (в левом нижнем углу экрана) и выберите пункт Завершение сеанса. Выведется диалог, предлагающий выключить или перезагрузить компьютер. После перезагрузки не забудьте вынуть DVD/CD, чтобы опять не загрузился SimplyMepis.

>>

Будьте как дома

Добро пожаловать в новую операционную систему SimplyMepis!

Установив и запустив систему, можете приступать к изучению рабочего стола. SimplyMepis использует окружение KDE – это дружелюбный и мощный рабочий стол, не бойтесь что-либо менять! Если вы запустили SimplyMepis прямо с диска-приложения и что-то идет не так, просто перезагрузитесь – с вашим ПК ничего не случится.



1 ЗВУК

Утилита *KMix* запускается автоматически при загрузке SimplyMepis. *KMix* – достаточно своеобразный регулятор громкости: на некоторых системах (в зависимости от оборудования) звук по умолчанию может быть еле слышным, поэтому здесь стоит проверить настройки.

2 НАСТРОЙКА

Центр ОС – главный инструмент конфигурирования. Вы обнаружите, что большинство вашего оборудования SimplyMepis уже настроил, но, возможно, некоторые настройки вам придется изменить – например, установить хитрый доступ к сети. При щелчке на инструментах слева появится соответствующая панель для внесения изменений.

3 ПРИЛОЖЕНИЯ

В отличие от некоторых дистрибутивов, SimplyMepis содержит много проприетарных программ, таких *RealPlayer*, видеокодеки, *Skype*, драйвера *NVIDIA* и другие.

4 МЕНЮ

K-Меню предназначено для запуска приложений – хотя SimplyMepis занимает всего один диск, приложений на нем множество. Выбирайте программы здесь.

5 ЯРЛЫКИ

Эти иконки запускают популярные приложения, например, *Firefox*. Совет: задержите курсор на любой из них – появится всплывающая подсказка о том, что она делает.

6 ПАНЕЛЬ ЗАДАЧ

Здесь отображаются работающие приложения, и между ними можно легко переключаться. По умолчанию SimplyMepis имеет два виртуальных рабочих стола – для переключения между ними щелкните на картинках (слева).

7 ПАНЕЛЬ

Панель KDE, прозванная Kicker, предоставляет легкий доступ к часто используемым инструментам. Часы настроены в соответствии с вашим системным временем.

8 СИСТЕМНЫЙ ЛОТОК

Системный лоток (tray) включает индикатор питания (на ноутбуках), ссылку на KMix и *KwikDisk* для быстрого доступа к смонтированным устройствам. Аквариум - это просто хранитель экрана.

9 ПОГОДА

Если вы подсоединены к Интернету, то этот сервис будет автоматически показывать погоду на родине SimplyMepis (Моргантаун, Западная Вирджиния). Если вы живете не там, щелкните правой кнопкой мыши для настройки местоположения.

ЧТО МОЖНО ДЕЛАТЬ В SIMPLYMEPIS

ИГРАТЬ В ИГРЫ



Как здорово нестись головой вниз с гигантской ледяной горы, собирая селедок! SimplyMepis содержит пару-тройку игр, включая родоначальника жанра Гонщик с планеты Пингвинов (*PlanetPenguin Racer*) – эта игра наверняка известна вам под названием Гонщик Tux (Tux Racer)! Имеется также несколько настольных и карточных игр, но программ можно добавить, установив SimplyMepis на жесткий диск.

БРОДИТЬ В СЕТИ

Иконка с глобусом в панели, расположенной по нижнему краю экрана, запускает *Firefox*, прекрасный web-браузер с открытым кодом. При наличии интернет-соединения хоть сейчас выходите в сеть – но если у вас WiFi с защищенным доступом или вы используете телефонный модем, сначала, возможно, придется запустить Центр ОС для настройки сети.

СЛУШАТЬ МУЗЫКУ И ПОДСОЕДИНИТЬ IPOD



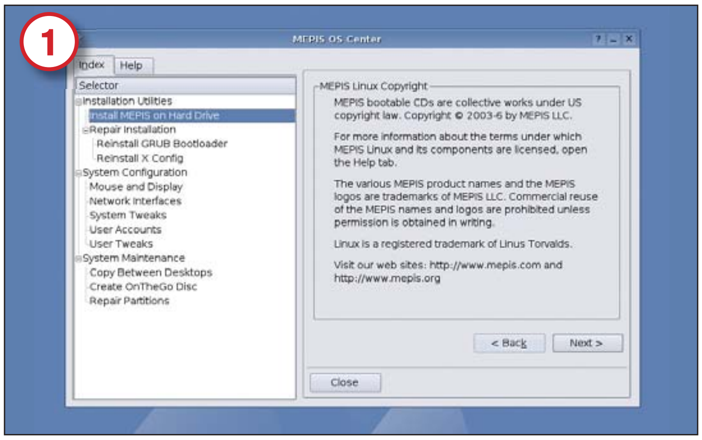
В SimplyMepis имеется *Amarok*, позволяющий проигрывать все типы музыкальных файлов, включая OGG, MP3, FLAC и потоковое аудио из Интернета. Можно даже управлять переносными аудио-устройствами типа iPod (правда, *Amarok* не умеет воспроизводить музыку, загруженную с музыкального архива Apple). Вы найдете *Amarok* в меню **Мультимедиа > Звук** (**Multimedia > Sound**).

ВЗЯТЬ В ДОРОГУ

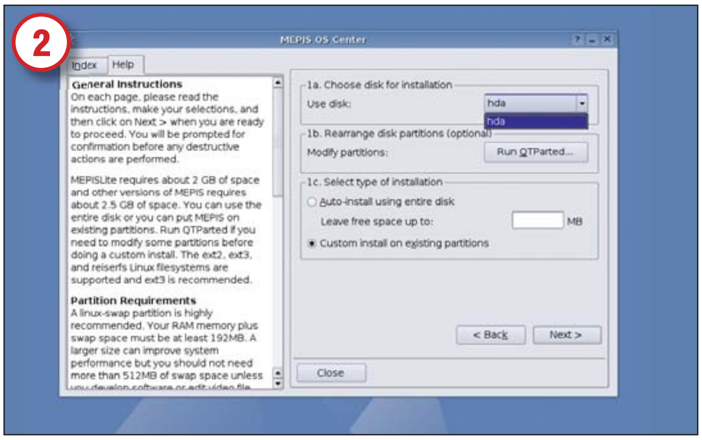
Пользователь с именем onthego задуман потому, что SimplyMepis имеет отличную функцию: создание Live-дистрибутива. Его можно взять его с собой куда угодно, прихватив еще и ваши файлы с рабочего стола. Чтоб задействовать функцию, нужен внешний USB-накопитель. В Центре ОС выберите из списка **Создать Мобильный Диск** (**Create OnTheGo Disc**). Выберите устройство, отформатируйте его и скопируйте туда файлы из вашего домашнего каталога. Когда захотите поработать с этим диском, подключите его и загрузитесь с SimplyMepis CD/DVD. Войдите в систему под именем **onthego** и с паролем 'onthego'.

ПРИДТИ И ОСТАТЬСЯ: УСТАНОВКА MEPIS НА ЖЕСТКИЙ ДИСК

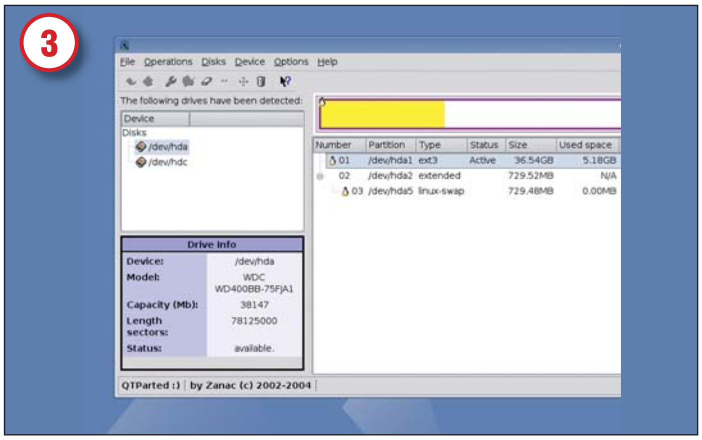
Поработав с Live CD, вы, возможно, захотите установить SimplyMepis на ваш жесткий диск: он будет работать гораздо быстрее, чем с DVD или CD, и вы сможете обновлять свой дистрибутив и устанавливать другие программы. Да и настройка «под себя» станет проще. Процедура установки несколько сложнее, чем запуск напрямую с DVD/CD, но если вы будете действовать по инструкции, все пройдет гладко. Если нужна помощь, обращайтесь на **www.mepis.org/forum**.



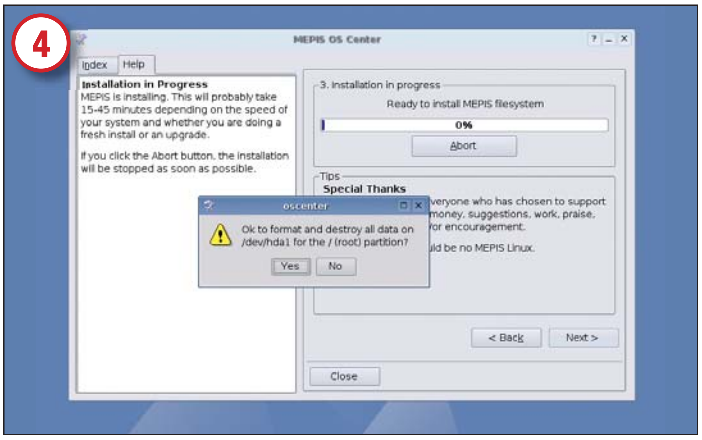
Щелкните на кнопке K-меню на рабочем столе SimplyMepis в KDE и выберите Mepis Config (Центр ОС). Будет запрошен пароль администратора – введите 'root'. В появившемся окне выберите слева Установить Mepis На Жесткий Диск (Install Mepis On Hard Drive) и щелкните на кнопке Далее (Next) справа.



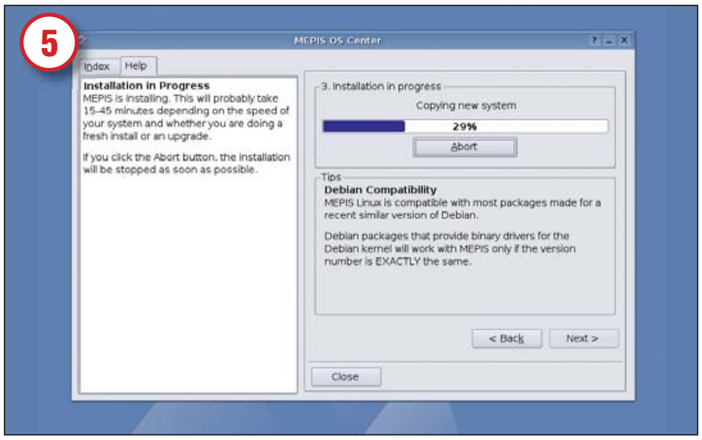
Это окно позволит вам выбрать жесткий диск, на который следует установить SimplyMepis: **hda** означает первый из них, а **hdb** – второй. Понадобится как минимум 3 ГБ дискового пространства. Если вы не хотите использовать весь диск (и грохнуть все, что там было!), щелкните на пункте Запустить QTParted (Run QTParted). В противном случае переходите к шагу 4.



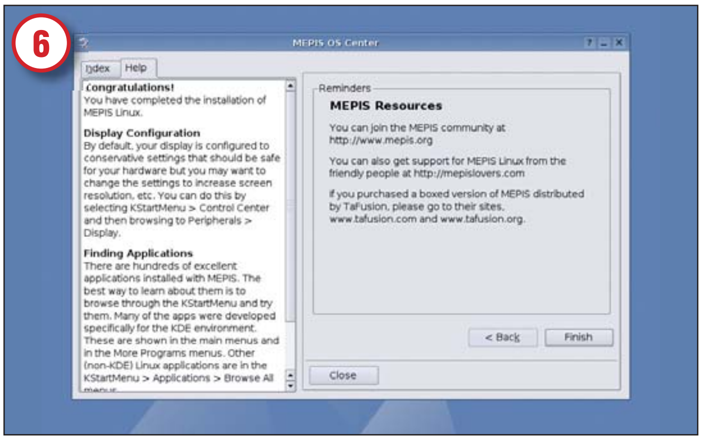
Для установки SimplyMepis на диск понадобится раздел размером не менее 3 ГБ с меткой '/' (корень файловой системы), отформатированный в файловой системе ext3, и swap-раздел размером 256–512 МБ. Это делается правой кнопкой мыши. Высвободить место можно, поджав некоторые разделы Windows, но сначала сделайте резервную копию! Нажмите Выполнить (Commit); по завершении, окно закройте.



Если на жестком диске есть важные данные, то, прежде чем перейти к этому шагу, создайте их резервную копию. Для полной очистки жесткого диска выберите Автоустановка На Весь Диск (Auto-Install Using Entire Disk); если вы произвели разметку вашего диска на третьем шаге, выберите Ручная Установка В Существующий Раздел (Custom Install On Existing Partitions). Проверьте настройки, нажмите Далее, и процесс пойдет!



Процесс установки займет от 15 минут до часа, так что можете попить чайку – или уж читайте советы, появляющиеся на экране. На очень быстрых машинах можно даже запустить в фоновом режиме другие приложения SimplyMepis, но тогда установка замедлится.



По завершении, нажмите Далее, чтобы установить загрузчик, затем назначьте имя пользователя и пароль. Если вас устраивают стандартные установки остальных параметров, то вы готовы к перезагрузке! Перезагрузите вашу машину, удалите DVD/CD, и SimplyMepis загрузится с вашего жесткого диска.

>>



Дорога к MEPIS

Уоррен Вудфорд (Warren Woodford), архитектор SimplyMepis, объясняет свой новый подход к Linux.

Linux Format: **Что заставило Вас начать работу над Mepis?**

Уоррен Вудфорд (УВ): Прежде всего, я искал платформу для личного использования, надежную и интуитивно понятную для меня. Первой в моем списке была Next OS, а второй – нечто от фирмы Xerox. Решая, на какой платформе я мог бы создать что-то новое и прекрасное, я подумал: «Это может быть Linux». И я дожидался выхода версии Linux, соответствующей моим минимальным требованиям. После выхода KDE 2.2 или около этого времени я сказал: «Эта оконная система обладает достаточным потенциалом», а консоль Linux уже обладала потенциалом.

Теперь – нужен кто-то для создания хорошего дистрибутива. И Red Hat казался худшим кандидатом, так как у них нет концепции «настольного компьютера» – добившись финансирования и известности, они сфокусировались на сервере, у сервера больший потенциал в смысле денег, и я не обвиняю их за это. Но вся их корпоративная культура разработана и построена вокруг идеи использования серверов. Они явно не имели хорошего понимания, чего хотят обычные пользователи. И я исключил их из уравнения.

LXF: **А как насчет SUSE?**

УВ: SUSE всегда казался мне незаконченным. Он какой-то художественный. Не то что он технически неправилен, просто он в дисгармонии со мной; он не кажется мне однородным, он не кажется интуитивно понятным для меня. Я чувствовал, что у него другая культура, и для меня это был не идеал. Так что я возложил все свои надежды на Mandrake. И продолжал возлагать надежды на Mandrake, но они никогда не делали то, что, как мне казалось, следует сделать. Им всегда чего-то нехватало, и нехватало из-за недостатка внимания к деталям или недостатка внимания к качеству.

В сентябре/октябре 2002 вышла новая версия Mandrake, и с ней имели место две фатальные ошибки, сказавшие мне, что Mandrake никогда не преодолеет недостатки своей культуры и не создаст продукт, отвечающий моим стандартам. Одной [из



TOP TRUMPS COURTESY: WINNING MOVES UK LTD

Визитка LXF

Уоррен Вудфорд

Родом из Западной Вирджинии, Уоррен работал в различных IT-проектах, а в 2002-м вложил все время и деньги в Mepis, взяв за логотип модный египетский иероглиф.

ВОЗРАСТ	57
НАЦИОНАЛЬНОСТЬ	Американец
СТАЖ РАБОТЫ В UNIX	4
ЯЗЫКОВ ПРОГРАММИРОВАНИЯ	15
КОЛИЧЕСТВО ПК	7
СУТОЧНАЯ НОРМА КОФЕ	ровно 1 чашка
ПАР САНДАЛИЙ	0

ПРЯМАЯ РЕЧЬ «Это просто факт из жизни, что американские покупатели предпочитают покупать американские продукты.»

ошибок] было то, что человек, отвечавший за формирование ISO-образов, три месяца спорил с сообществом тестеров: действительно ли какой-то из ISO-образов слишком велик, чтобы его можно было прожечь на большинстве компьютеров. Образ просто был на пределе стандартного CD емкостью 700 МБ, и этот парень высокомерно заявлял: «Вы не правы, а прав я, все вмещается». Но неправ был он сам. Множество людей сообщили, что не могут прожечь ISO-образы.

О ДИСТРИБУТИВЕ LINUX КАК О ПРОДУКТЕ:

> «Основная трудностью любого дистрибутива – баланс между нуждами больших коммерческих потребителей и старанием остаться в сообществе.»

Другая ошибка – после четырех бета-версий и, я уверен, четырех релиз-кандидатов, за два дня до выпуска финальной версии они изменили *automount* и вывели его из строя более чем на два месяца, а сами тем временем спорили с сообществом тестеров, знает ли сообщество, о чем говорит, когда сообщает о проблеме.

Вот в это время, я точно помню – 21 ноября 2002, я и принял решение создать свой собственный дистрибутив. Я не знал, в каком направлении пойдет развитие и что получится. Я поделился этим с несколькими людьми, и все они назвали меня сумасшедшим, потому что и без меня были сотни различных дистрибутивов. Но на тот момент у меня не было выбора.

LXF: **Что оттолкнуло вас от работы с Mandrake?**

УВ: По большей части это был весь мой опыт работы в Силиконовой Долине, где я работал консультантом, приходя и налаживая технологические подразделения, неспособные выпустить продукт – [фирмы] имели фатальные ошибки в своей инженерной философии. Я видел, что Mandrake испытывает те же трудности, что и фирмы, с которыми я работал до этого. Я чувствовал, что никогда не ликвидирую эти проблемы. Вот я и создал свой собственный дистрибутив, потому что решил: «Для моего дистрибутива есть место, никто не делает то, что хочу сделать я, и может быть, другим людям это тоже понравится. Попробую и посмотрю, что получится.»

LXF: **А откуда взялось имя Mepis?**

УВ: Ну, как-то мы с другом хотели создать софтверную фирму в Париже, и по поводу названия у нас возникло недопонимание, так как я не мог понять его очень сильный французский акцент. Я подумал, что он сказал Mepis [Уоррен произносит это как «Меп-с»], а я не понял, но тут обнаружил, что Mepis является комбинацией пяти букв, которая еще нигде не использовалась и потому была невероятно хорошим названием для продукта. Я ухватился за это имя и остановился на нем, и когда пришло время, я сказал: «OK, у меня нет времени на маркетинг, я хочу просто сделать это», – и я вновь использовал имя Mepis [теперь он произносит «М-пис»].

Я осознавал, что Mepis можно использовать как аббревиатуру. Например, сегодня это может быть Моя Экстраординарная Персональная Информационная Система. Вероятно, оно никогда не будет означать что-либо, но вполне может.

LXF: **Итак, вы взяли Mandrake и хотели его исправить?**

УВ: Нет, начал я с Debian. Я был под большим впечатлением от LiveCD [Knoppix] Клауса Кноппера (Klaus Knopper). Мне казалось, что никто не осознает возможности LiveCD; о его назначении думали очень узко. Итак, для начала я создал дистрибутив на базе Debian с заимствованиями из работы Клауса. Я создал версию Linux, в которой вы могли делать все что угодно с LiveCD, как если бы он был установлен на жестком диске. Не ограниченный, не модифицированный, а абсолютно стандартный запуск, абсолютно стандартный рабочий стол.

И ух ты, что если создать и установщик как обычное приложение? Прекратить практику создания установщика как специального режима загрузки системы. Запуск LiveCD – это уже нечто новое. Сначала вы проводите тест-драйв Linux, то есть пробуете его, и он работает так, как работал бы после установки. Очевидно, что некоторые вещи работать не будут, так как нет файловой системы с возможностью записи, но вы можете испробовать многое из доступного. Итак, LiveCD – это прекрасно, но мы не называем наш диск LiveCD: мы называем его загрузочным CD для испытания и принятия решения, потому что некоторые люди, пытаясь сообразить, что такое Mepis, классифицируют его как LiveCD и упускают множество того, что он реально предоставляет.

Второй момент: почему бы не иметь графический установщик, запускать его с LiveCD и устанавливать Mepis, если вы убедились, что он вам подходит? Мы сделали это. Третий момент: уж если имеется такое графическое окружение, почему бы не добавить к имеющемуся набору несколько инструментов, необходимых для обслуживания системы, восстановления при ошибках? Например, вы – любитель, запутались в настройках файловой системы и больше не можете загрузиться. Почему бы не иметь возможность загрузиться с CD и запустить небольшую утилиту, которая переустановит для вас X? Мы включили инструменты подобного типа.

LXF: **Когда вы выбрали Debian, вас заботило, что и другие поставщики разрабатывали его и потерпели неудачу?**

УВ: Я думаю, что основной трудностью для любого дистрибутива является балансирование между нуждами больших коммерческих потребителей и попытками удержаться в сообществе разработчиков Open Source, от которого все зависят и которому все благодарны. Некоторые [члены сообщества] словно живут в параллельной вселенной – они считают, что все должны вести себя так, как будто сейчас XXIV век и у всех все есть, так что можно просто отдать все планете и все будет прекрасно.

>>

Но реальность такова, что люди, посвящающие свое время Open Source, должны кушать и иметь крышу над головой, так что подход полного социализма *Star Trek* – это прекрасный идеал, но не реальность. Есть такие люди, а есть прагматики, считающие «Эй, люди должны зарабатывать себе на жизнь». Мы прикладываем все усилия, чтобы отметить вклад тех, кто выполняет работу по личным причинам, и стараемся не обидеть и сообщество.

LXF: **Как вы думаете, Mepis приживется на рынке среди таких соперников, как Ubuntu?**

УВ: Похоже, что Ubuntu все взял от Debian и поставил на этом свое имя, а также нанял несколько людей с намерением сделать ответвление от Debian. Об этом вряд ли есть хоть какие-то официальные заявления Ubuntu, но мне это совершенно ясно; их подход в том, чтобы сперва взять все что можно из Debian, налепить свой ярлык и мощным маркетингом вывести дистрибутив на арену... затем попытаться вернуться назад и сделать что-нибудь уникальное на закуску, чтобы можно было сказать: «Это действительно наше, а не унаследованное от Debian». И мне кажется, это одна из лучших стратегий большого бизнеса. Я думаю, есть пуристы, которым это не нравится, но хороший, расчетливый и твердый бизнес вполне имеет смысл. «Эй, здесь что-то дают, ну так возьмем. Подпишем свое имя, это позволено. Вложим много денег в маркетинг и получим большое признание нашей марки». Тут есть логика.

В данный момент, Ubuntu и Mepis никак не сталкиваются друг с другом, разве что Ubuntu налаживает мосты к Hewlett Packard в Европе, и я не знаю, будем мы соперничать с ним в этом или нет. А так мы вообще не сталкиваемся друг с другом. Я видел только одного человека во всем мире, которому нравится Ubuntu, и тот сказал, что не взял бы его для своей фирмы, он нравится ему лично.



Как серверный вариант, Ubuntu представляет слишком много около-Debian философии вроде «Из этого супер-универсального меню вы можете получить все на свете». Наш серверный продукт [SoHo Server] похож на наш настольный продукт: установите его, и все – можете работать. Что хорошо и что плохо, мы решим за вас.

А если вы пришли с Windows, вам это и нужно. Зачем становиться экспертом в Linux, чтобы знать, что установить? Нужна хорошая и легкая отправная точка, что-нибудь, с чем вы сможете работать прямо сейчас. Затем вы можете потихоньку изучать что захотите, и настроите систему, как вам нравится. LXF



Представление начинается

Наверняка вы уже прикупили себе один из тех широко разрекламированных домашних кинотеатров. Или еще нет? **Сергей Яремчук** расскажет, как превратить обычный дистрибутив Linux в домашнюю мультимедиа-станцию.

Бытует мнение, что Linux – невероятно сложная в освоении система, от которой трудно ожидать продвинутой работы с мультимедиа. Все это уже в далеком далеком прошлом. За последние годы появились дистрибутивы, максимально дружелюбные к пользователю, а качество приложений предназначенных для просмотра видео, телевизионных программ, изображений и прочего, не только не уступает таковым для Windows, но и в некоторых случаях даже превосходит их. Кроме того, появились специализированные программы (и даже специализированные дистрибутивы), позволяющие объединить все утилиты и сделать из компьютера единый медиа-центр. Среди них стоит отметить китайский Linpus(tm) Media Center (**http://www.linpus.com.tw/Solutions/Embedded/Media_Center/default.htm**), внешний вид которого напоминает продукт компании Microsoft, а также два LiveCD: *Freevo* LiveCD (**http://www.pegasos.org/downloads/torrents/Freevo-live.torrent**) и Wolvix Media Edition (**http://wolvix.org/**). В состав последнего входит установленный и настроенный *Freevo*, о котором мы сегодня и поговорим.

Проект Freevo

Freevo (**http://Freevo.sourceforge.net/**) является открытым домашним медиа-центром, работающим на компьютере под управлением Linux и базирующимся на других свободных разработках. Среди его возможностей: просмотр любых видео, аудио и графических файлов, расположенных на жестком диске компьютера, а также доступных через сеть. Имеется возможность просмотра и сохранения на диск телевизионных и спутниковых программ. Программа позволяет читать новости и просматривать программу телепередач прямо из Интернета. Сигнал может выводиться как на телевизор, через TV-выход графической карты, так и на монитор. Управление осуществляется традиционно (при помощи мыши и клавиатуры), а также с пульта дистанционного управления.

Freevo разрабатывается Кристером Лагерстрёмом (Krister Lagerström) из Канады. Первая версия, выпущенная в мае 2002 года, представляла собой синий экран с простой системой меню. Сейчас это не просто программа, а полноценная платформа для организации домашнего медиацентра, а поддержка подключаемых модулей позволяет существенно расширить возможности и даже превзойти известные аппаратные решения. Имеющиеся сейчас модули умеют не только изменять внешний вид системы, но и кодировать аудио и видео, записывать CD и DVD-диски, отправлять и принимать почту, интегрироваться с iPod, получать погоду и многое другое. Написан *Freevo* на языке высокого уровня *Python*.

Установка

Программа легко устанавливается, не требует компиляции и не имеет большого числа внешних зависимостей, но вот для настройки необходимо иметь достаточные знания, так как документация рассчитана явно на подготовленного пользователя, хорошо знающего английский язык. На сайте проекта доступны исходные тексты, на странице About перечислены основные зависимости: интерпретатор *Python*, библиотеки SDL, *Pygame*, *Python Imaging Library* и *Python Twisted*, а также *Mplayer* и *Xine*. Сразу бросаться и скачивать их я бы не рекомендовал, так как большая часть, скорее всего, уже установлена в вашей системе. Инсталляционный сценарий пожалуется, если ему чего-то не хватает, либо порекомендует обновить те или иные программы, например:

```
# python setup.py install
checking for mmpython... not found
please download it from http://www.sf.net/projects/mmpython and install it
```

Скачайте указанный пакет (или найдите его в составе вашего дистрибутива), установите и повторите попытку. Обратите внимание, что для установки потребуется права суперпользователя. *Freevo* можно найти в портах Gentoo Linux, репозитарии Debian, доступны rpm-пакеты для SUSE Linux, Mandriva, Fedora Core и других дистрибутивов. Есть *Freevo* и в Сизифе (ALT Linux), поэтому не стесняйтесь использовать свой пакетный менеджер. Кроме всего прочего, для реализации тех или иных возможностей требуется установить сопутствующие

библиотеки и приложения: *Mplayer* или *Xine*, *LIRC* и *PyLIRC*, *PyGame*, *xmltv*, *jpgtran*, библиотеки для работы с DVD и другие.

Настройка

Все настройки *Freevo* хранятся в трех конфигурационных файлах: **Freevo.conf**, **Freevo_config.py** и **local_conf.py**. Файл *Freevo*.conf генерируется автоматически во время установки, находится в **/etc/Freevo** и содержит основные параметры вашей системы, разрешение экрана и пути к используемым файлам. Второй файл – **Freevo_config.py** находится в **/usr/share/Freevo** и содержит настройки *Freevo*, используемые по умолчанию. Изменять его не рекомендуется – переопределить параметры можно в файле **~/.Freevo/local_conf.py**. Создавайте его из **Freevo_config.py**.

```
$ cd; mkdir .Freevo
$ cp /usr/share/Freevo/Freevo_config.py
~/.Freevo/local_conf.py
```

Теперь можно приступать к редактированию. Файл содержит несколько секций, некоторые из них можно пока не трогать (более детально все настройки описаны на странице DetailedConfig в документации *Freevo*). Самое главное: перед параметрами не должно быть никаких пробелов или табуляций, иначе вы получите сообщение ошибке синтаксиса, хотя внешне все будет выглядеть нормально. [хотите знать, почему? Читайте «Уроки Python» Серегя Супрунова! – прим. ред.]

Секция General freevo settings

Первая секция содержит общие установки. Например, строка **DEFAULT_VOLUME = 40** указывает на уровень громкости, выставляемый по умолчанию. Изменив значение параметра **START_FULLSCREEN_X = 0** на **1**, можно заставить *Freevo* запускаться в полноэкранном режиме. Используйте этот параметр, только когда полностью настроите пакет! Строка **ROM_DRIVES = None** активирует автоматическое обнаружение всех CD или DVD устройств с использованием **/etc/fstab**. Для отключения этой возможности используйте пустые скобки []. В некоторых случаях придется прописать путь вручную.

```
ROM_DRIVES = [ ('/mnt/cdrom', '/dev/
cdrom', 'CD-ROM') ]
```

Чтобы шум привода не мешал при просмотре фильма, ограничьте скорость CD-ROM до 8x.

```
ROM_SPEED =8
```

И позвольте выключать компьютер при помощи меню *Freevo*.

```
ENABLE_SHUTDOWN_SYS = 1
```

Также в этой секции традиционно переопределяются события (events), которые описаны в файле **src/event.py**. Например, чтобы во время просмотра фильма увеличить контрастность до 100 нажатием на клавишу 1, нужно прописать следующее правило.

```
EVENTS['video']['1'] = Event(VIDEO_SEND_
MPLAYER_CMD, arg='contrast -100')
```

Подключаемые модули

В следующей секции настраиваются подключаемые модули *Freevo*. Все установленные модули автоматически запускаются из файла **Freevo_config.py**. Например, строка для запуска модуля TV выглядит так.

```
plugin.activate('tv', level=10)
```

В пользовательском файле, как правило, либо отключают запуск ненужного модуля, либо переопределяют его параметры. Если у вас (как и у меня) нет TV-тюнера и джойстика, добавьте следующие строки.

```
plugin.remove('tv')
plugin.remove('joy')
```

Каталоги

Эта секция позволяет изменить порядок сортировки файлов в каталогах, включить «автозапуск» для содержимого каталогов и вывод информационных тэгов. Например, чтобы разрешить автовоспроизведение для музыкальных файлов и изображений, используйте:

```
DIRECTORY_AUTOPLAY_ITEMS = [ 'audio',
'image' ]
```

Позвольте создавать для них список воспроизведения (playlist).

```
DIRECTORY_CREATE_PLAYLIST = [ 'audio',
'image' ]
```

И добавьте его в каталог

```
DIRECTORY_ADD_PLAYLIST_FILES = [
'audio', 'image' ]
```

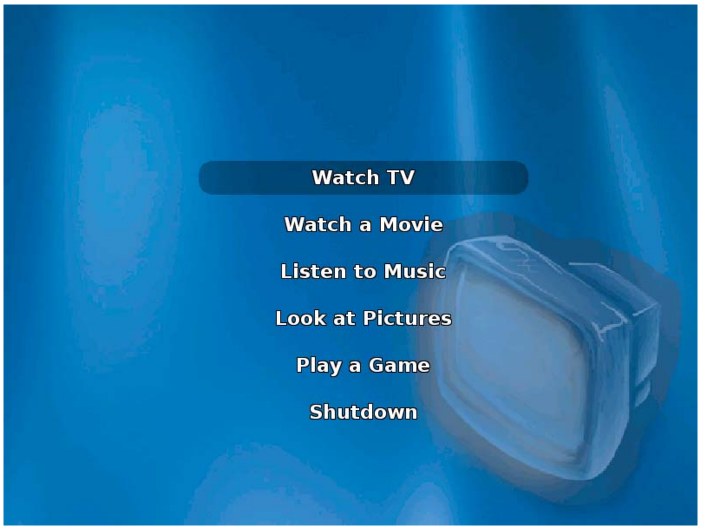
Для большего удобства можно создать в каталоге файл **folder.fxd**. Он имеет приблизительно такую структуру:

```
<Freevo>
 <folder title="Заголовок каталога" img-
cover="рисунок.png">
 <setvar name="directory_autoplay_single_
item" val="0"/>
 <info>
 <content>Краткое описание каталога</
content>
 </info>
 </folder>
</Freevo>
```

Можно использовать один общий файл **folder.fxd** (например, в домашнем каталоге) и ссылаться на рабочие директории следующим образом:

```
<directory recursive="1">/mnt/mp3/party/
rock</directory>
```

Изображения должны быть сохранены в формате JPEG или PNG и иметь размер 280x200 для фильмов, 200x200 для аудио и 200x160 для каталогов с рисунками. Теперь,



Меню Freevo.

если указать этот каталог *Freevo*, в меню будет выведен заголовок и рядом соответствующий ему рисунок. Кроме того, *Freevo* позволяет защитить некоторые каталоги от просмотра. Для этого в них необходимо поместить файл .password, в котором указать пароль для доступа. Это защита не спасет при наличии доступа к консоли, но при управлении с пульта ДУ ее вполне достаточно. Пароль желательно использовать цифровой, так как в этом случае его можно будет легко ввести с пульта.

Настройка каталогов видео, аудио, изображений и игр

По умолчанию в меню для просмотра фильмов, изображений и музыки отображается домашний каталог пользователя, корневой каталог и смонтированные сменные носители. Удобнее указать каталоги, в которых хранится требуемая информация, вручную, чтобы сократить время поиска. Для этого используются такие директивы:

```
VIDEO_ITEMS = [ ('Movie', '/mnt/media/
movies'),
 ('Klips', '/mnt/media/klips') ]
```

В качестве параметров могут выступать как каталоги, так и *fxd* файлы. При помощи следующей директивы можно указать используемый видеопроигрыватель. Для *Mplayer*:

```
VIDEO_PREFERED_PLAYER = 'mplayer'
```

Для XINE:

```
VIDEO_PREFERED_PLAYER = 'xine'
```

Аналогично для каталогов изображений и музыки.

```
AUDIO_ITEMS = [ ('Music', '/mnt/media/
music'),
 'fxd/webradio.fxd' ]
```

Вторая строка указывает на файл, в котором прописаны Интернет-станции потокового аудио. Готовый файл идет в комплекте с *Freevo*, при необходимости его можно подредактировать, добавив свои любимые станции.

```
IMAGE_ITEMS = [ ('Foto', '/mnt/media/foto'),
('Pics', '/mnt/media/pics') ]
```

Для игр необходимо указать не только заголовок и каталог, но и путь к исполняемому файлу эмулятора, аргументы и, по возможности, рисунок.

```
GAMES_ITEMS = [ ('SUPER NINTENDO', '/
home/media/games/snes/roms',
('SNES', '/usr/bin/zsnes', '-m -r 3 -k 100 -cs -
u', '', None )) ]
```

Прочие настройки

Freevo поддерживает визуальные темы (skins). В комплекте поставляются три: >>

ИСПРАВЛЯЕМ АКЦЕНТ

К сожалению, при настройках по умолчанию *Freevo* плохо справляется с выводом локализованного текста, поэтому в меню желательно использовать только английские буквы. Что еще печальнее, данная ситуация сохранится, пока не будет произведен переход на Unicode всех приложений. Проблему можно решить (по крайней мере, частично), заменив используемые по умолчанию TTF-шрифты, хранящиеся в каталоге **/usr/share/freevo/fonts/**, на **mscorefonts** (в случае Unicode), либо другие поддерживающие системную кодировку. В локализованных дистрибутивах вроде ALTLinux или ASPLinux, также есть готовые шрифты, которые можно использовать с Freevo. Для полной поддержки русских шрифтов необходимо наложить исправления и пересобрать PyGame. Патч можно загрузить по адресу: **http://m0sia.ru/files/utf8.patch**



Рис. 2. Предосмотр изображений при помощи *Freevo*.

Blurr, Info и Noia, на сайте проекта можно найти еще пять готовых тем.

```
SKIN_XML_FILE = 'blurr'
```

Темы представляют собой XML-файлы (с расширением fxd), на основании которых можно создавать собственные вариации. Далее в файле идет описание вывода меню, установки *xine*, *mplayer*, просмотра TV и локаль.

```
LOCALE= 'ru_RU.UTF-8'
```

Практически последним пунктом идут параметры управления *Freevo*. Об этом мы и поговорим подробнее

Управление

Управление является одной из сильных сторон *Freevo*. Кроме стандартных клавиатуры и мыши, можно использовать джойстик. Для управления можно использовать встроенный web-сервер, который позволяет не только просматривать локальные данные, но и производить запись web-трансляций вручную и по расписанию. С помощью web-браузера можно также просматривать файлы помощи. Для запуска web-сервера отредактируйте следующие строки:

```
plugin.activate('www')
WWW_PORT = 8080
```

И установите пользователя и пароль для доступа к серверу.

```
WWW_USERS = { 'user': 'password' }
```

Теперь наберите в web-браузере http://localhost:8080/, зарегистрируйтесь и приступайте к работе (рис. 3). Точно таким же образом на компьютер можно зайти удаленно.

Особенно полезна возможность дистанционного управления *Freevo*. Для этого можно использовать ИК-пульт через (LIRC) или телефон с поддержкой Bluetooth.

Через Bluetooth

Эта возможность является предпочтительной: радиосигнал хорошо распространяется по всей квартире, так что компьютер может стоять даже в соседней комнате. Пользоваться пультом ДУ в такой ситуации не совсем удобно. Для реализации данной возможности нам потребуется модуль расширения Bluetooth, который можно скачать со страницы **Plugins** проекта *Freevo*. Его установка не вызывает затруднений:

```
$ tar xvf bluetooth.tar
# python setup.py install
```

Ядро большинства дистрибутивов уже поддерживает Bluetooth, и в комплекте имеются необходимые утилиты. Подобно на работе с USB и Bluetooth в Linux останавливаться не буду – читайте статью «make menuconfig: готовые рецепты» в *LXF78*. Информацию по настройке демона *hcid* и подключению телефона можно найти во врезке.

Добавьте в **local_conf.py** следующие строки:

```
Активируйте модуль.
plugin.activate('blue')
```

Укажите на связанное устройство (в некоторых дистрибутивах это может быть, например, **/dev/ rfcomm0**).

```
BLUE_RFCOMM = "/dev/bluetooth/
rfcomm/1"
```

Переменная для установки значения времени опроса (0 – опрос постоянно, 10 – слишком долго).

```
BLUE_SLEEP = 1
```

Проверка соединения.

```
BLUE_KEEP_ALIVE = 100
```

Далее запишите соответствие кнопок телефона командам *Freevo*. Проверить, какая кнопка нажата на телефоне, можно при помощи команды **Freevo bluebutton** либо **Freevo bluetooth_config**:

```
BLUE_CMDS = {
'1' : 'PLAY',
'2' : 'VOL+',
'3' : 'VOL-',
'4' : 'LEFT',
'5'          : 'UP',
'6'          : 'RIGHT',
'7'          : 'SELECT',
'8' : 'DOWN',
'9' : 'EXIT',
'0' : ' UNILOCK ' }
```

Команда UNILOCK позволяет временно блокировать и разблокировать клавиатуру телефона, например, при ответе на звонок.

Настройка телевизора

Последним шагом будет вывод изображения на телевизор. Для этого необходимо добавить запись о новом устройстве в файл **/etc/X11/xorg.conf**. Формат записи будет зависеть от используемой видеокарты. За дополнительной информацией попробуйте обратиться на сайт производителя или разработчика драйвера. Если ничего из перечисленного не помогло – напишите вопрос на наш форум www.linuxforum.ru или в рубрику «Вопрос? Ответ!». Приятного просмотра! **LXF**

СВЯЗЬ ЧЕРЕЗ BLUETOOTH

Вставьте USB-Bluetooth адаптер и проверьте его обнаружение при помощи команды:

```
#lsmod | grep usb
hci_usb 14216 0
bluetooth 44292 1 hci_usb
```

Кроме того соответствующая информация должна появиться в **/var/log/messages**.

```
#cat /var/log/messages
….
Mar 11 21:23:37 (none) kernel: usb 1-1: new full speed USB device using uhci_hcd and address 2
Mar 11 21:23:44 (none) kernel: Bluetooth: Core ver 2.7
Mar 11 21:23:44 (none) kernel: NET: Registered protocol family 31
Mar 11 21:23:44 (none) kernel: Bluetooth: HCI device and connection manager initialized
Mar 11 21:23:44 (none) kernel: Bluetooth: HCI socket layer initialized
Mar 11 21:23:44 (none) kernel: Bluetooth: HCI USB driver ver 2.8
Mar 11 21:23:44 (none) kernel: usbcore: registered new driver hci_usb
```

Затем отредактируйте файл настройки демона hcid (**/etc/bluetooth/hcid.conf**)

```
options {
autoinit yes;
security auto;
pairing multi;
pin_helper /etc/bluetooth/bluepin;
}
device {
name "freevo";
class 0x100;
pkt_type DH1,DM1,HV1;
iscan enable;
pscan enable;
lm master;
lp hold,sniff,park;
auth enable;
encrypt disable;
}
```

Создайте файл **/etc/bluetooth/bluepin**, который будет содержать PIN для доступа к телефону.

```
#!/bin/sh
echo "PIN:12345"
```

Сохраните его и сделайте исполняемым:

```
$ chmod 700 bluepin && chown sergej bluepin
```

Запустите демон hcid.

```
#hciconfig hci0 up; hcid -f /etc/bluetooth/hcid.conf"
```

Проверьте, обнаружено ли Bluetooth устройство:

```
#hciconfig –a
```

Просканируйте сеть в поисках телефона:

```
# hcitool scan
Scanning ...
 00:0F:DE:1A:34:05 T630
```

И свяжите канал:

```
#rfcomm bind rfcomm0 1"
```

Проверьте, все ли сделано правильно:

```
#rfcomm show
rfcomm0: 00:0F:DE:1A:34:05 channel 1 clean
```

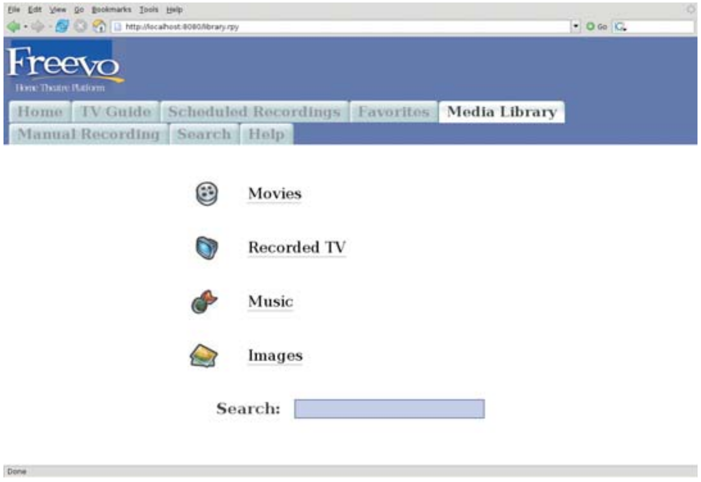


Рис. 3. Freevo можно управлять через web-браузер.



СИЛА СОЛНЦА

Конкуренция в мире Open Source обостряется: фирма Sun открыла исходные коды своей Unix-системы уровня предприятия. Майк Сондерс исследовал OpenSolaris и покажет вам, как с ним работать.

Вторник, 14 июня 2005 года, можно считать одним из значительнейших дней в истории Open Source. В этот день Sun Microsystems открыла миру исходный код и процессы разработки своей первоклассной операционной системы Solaris из семейства Unix, под именем OpenSolaris. Пять лет назад такой ход со стороны крупной коммерческой Unix-фирмы был бы немыслимым: ведь исходный код – главный козырь подобных фирм, и сама идея вовлечения посторонних в процесс разработки считалась ошибочным шагом. Подобные изменения демонстрируют, насколько движение Open Source и свободное ПО влияют на большой бизнес и как он реагирует.

Предназначенный для промышленности, Solaris не слишком популярен среди обычных пользователей, но является основой ряда важнейших компьютерных систем мира, обслуживая критические базы данных, крупномасштабные web-серверы и центры обработки данных таких фирм-гигантов, как Boeing, Viacom и Bank of America. Исторически Sun фокусировалась на собственной платформе SPARC для Solaris, но недавно фирма обнаружила, что соотношение «цена–производительность» ПК на x86 – слишком сильный аргумент для любого покупателя, чтобы его игнорировать, и, как следствие, поспешила с разработкой Solaris для обычных ПК.

Так что же подвигло Sun на столь дерзкий, можно сказать, исторический шаг – открыть свою флагманскую ОС? Sun «трубит» о своей истории поддержки открытых стандартов и выражает надежду, что открытие будет «способствовать инновациям и сотрудничеству, необходимым для формирования новых сообществ разработчиков и пользователей». Всё правда, но Sun, несомненно, еще и чувствует накал популярности Linux на рынке малых и средних серверов. Система Solaris заслужила репутацию надёжной и быстрой на многопроцессорных серверах – зачастую именно эту ОС выбирают для хранения и обработки больших объёмов данных, тогда как Linux считается сравнительно незрелым и неустойчивым. Но эта ситуация не вечна, и Sun понимает, что перемены уже витают в воздухе.

Пословица Open Source «больше глаз, меньше ошибок» тоже относится к критическим факторам. У коммерческого Solaris есть огромное сообщество разработки и поддержки, но вовлечение ещё большего числа людей – мудрый шаг Sun, нацеленный на сокращение времени решения проблем и реализации функций в ОС. Предоставив миру Solaris под именем OpenSolaris (под своей одобренной OSI лицензией CDDL), Sun сможет воспользоваться преимуществами Open Source, сделав ряд своих высокотехнологичных функций доступными для всех и получив почет в ответ.

На следующих страницах мы расскажем вам, как поставить и запустить эту увлекательную систему, изучим её основные функции и «нырнём» поглубже. На прилагаемом диске вы найдете все необходимое, чтобы испытать эти знания на практике!

ЗАЧЕМ ПРОБОВАТЬ OPENSOLARIS?

- Расширьте свои навыки работы в Unix – Solaris абсолютно независим от Linux.
- Попробуйте функции класса предприятия на домашней системе.
- Разрабатывайте и тестируйте программы на совместимость с коммерческим Solaris.
- Поглядите на основного конкурента, с которым сталкивается Linux-сообщество.

Запускаем OpenSolaris

Во-первых, OpenSolaris включает не всё, что есть в Solaris. Sun разрабатывает свою систему (бывшую SunOS) с 1980-х годов, и вложила колоссальные средства в высокотехнологичные функции, обеспечивающие потрясающую масштабируемость и надёжность, зачастую используя код, разработанный другими компаниями. Следовательно, по юридическим соображениям нельзя открыть всё разом, так что Sun выпустила пока «сердце» своей ОС, с надеждой позже добавить остальное. Это привело к появлению ряда «дистрибутивов» OpenSolaris, взявших открытый фирмой код и расширивших его до пригодной к использованию ОС, включив существующие свободные инструменты.

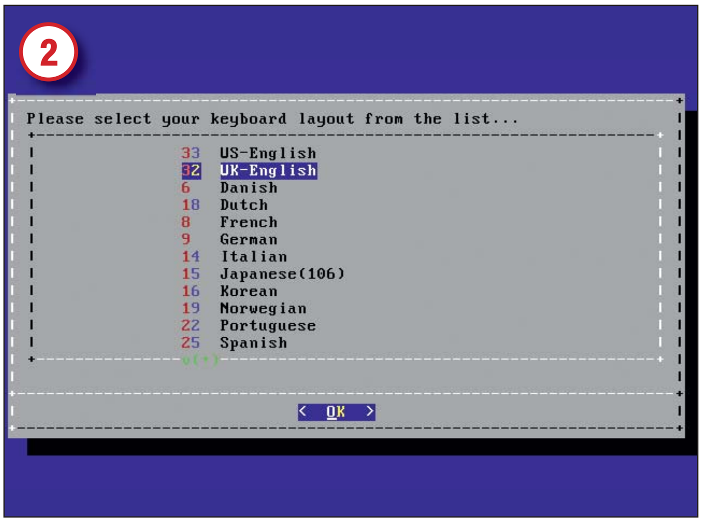
Система Belenix Live (**http://belenix.sarovar.org**), помещённая на наш диск, одна из таких ОС. Belenix использует ядро Solaris, библиотеки и «пространство пользователя» – userland (основные инструменты командной строки), предоставленные сообществу фирмой Sun, с более привычными дополнениями (GNU и Open Source), увеличивающими общую полезность.

Среди других ОС, базирующихся на OpenSolaris – Nexenta, пытающаяся скрестить Solaris и Debian, и Schillix, представляющий собой компактный вариант. Будучи системой Unix, Solaris использует *X Window System*, как и Linux, и Belenix включает *X.org* 6.9.0 вместе с *Xfce* и KDE.

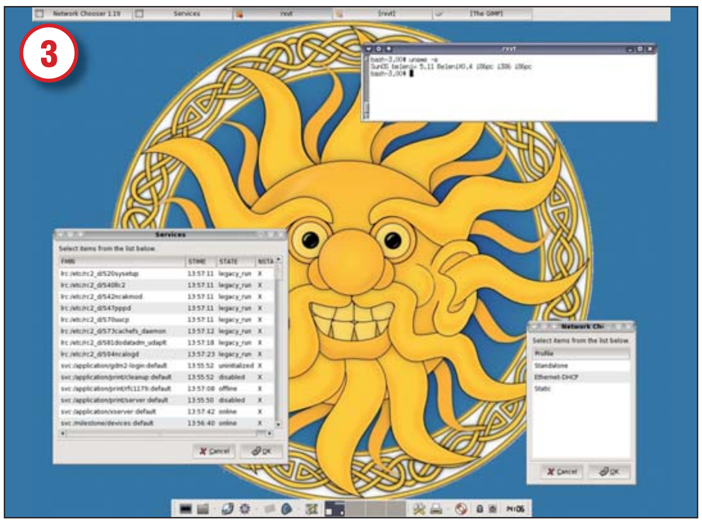
Внешне Belenix похож на многие дистрибутивы Linux. Solaris стремился к этому ещё до того как Sun открыл его код, так что Solaris 10 (текущий релиз) использует в качестве рабочего стола по умолчанию основанный на Gnome JDS. Но, копнув поглубже, вы найдете поразительно мощные функции и возможности, благодаря которым Solaris ходит в любимчиках у многих администраторов. Здесь нет функций, способных совершить революцию в работе с настольными системами, но на «большом железе» они могут оказаться очень ценными. Со многими из них мы познакомимся через мгновение…



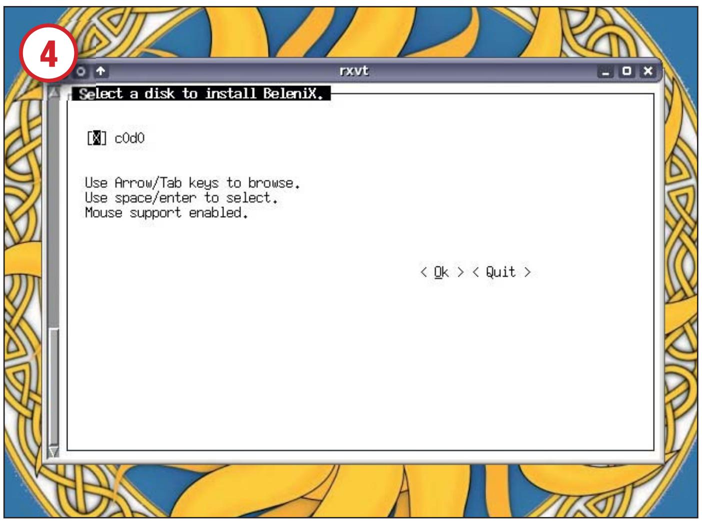
Вставьте DVD в привод, перезагрузите компьютер, и – вперёд. Belenix – Live-версия OpenSolaris, так что работает непосредственно с диска и не требует установки: просто выберите её в загрузочном меню, нажмите Enter, и она запустится (требуется 256 МБ памяти). Опыт работы в Linux/Unix не помешает!



Теперь система готова к загрузке. Если вы увидите серьёзные ошибки во время процесса запуска, попробуйте перегрузиться и выбрать пункт «No ACPI» на первом шаге. Нужно будет выбрать раскладку клавиатуры и рабочий стол; пока что выберите *Xfce*, поскольку он быстр, а его меню содержит ссылки на полезные инструменты Solaris.



Если всё будет нормально, вы окажетесь в графическом окружении – Solaris под капотом, популярный GUI снаружи (KDE также доступен и великолепно запускается). Вы обнаружите множество знакомых приложений, но чтобы увидеть внутренности OpenSolaris, откройте терминал, щёлкнув левую иконку на нижней панели.



При желании вы можете установить Belenix на ваш жёсткий диск, хотя это требует технической подготовки и выходит за пределы этой статьи. Однако если вы всё же хотите сделать это, ознакомьтесь с **hdinstall.html** в разделе Distros/Belenix на нашем диске, затем введите **hdinstaller** в окне терминала…

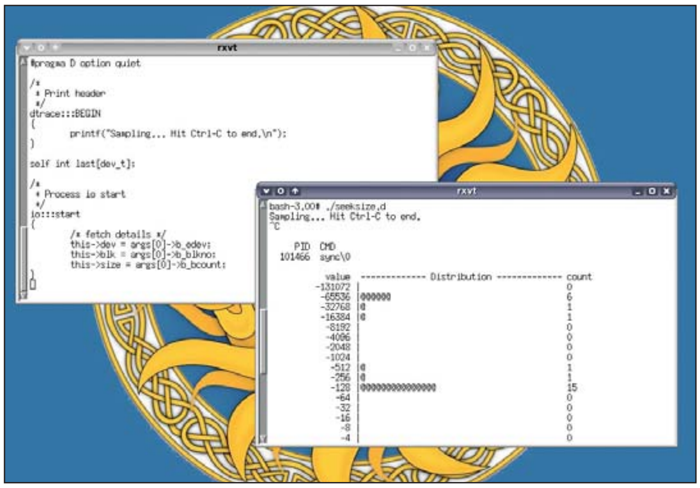
>>

« Принимайтесь за вашу новую ОС

Загрузившись, приступим к изучению системы! Если вы регулярно работаете в Linux, файловая система и многие привычные команды будут вам знакомы – Unix есть Unix, как его ни пиши. К счастью, Belenix использует командную оболочку *GNU Bash*; однако вы обнаружите, что функциональность многих инструментов командной строки, используемых в Solaris, довольно ограниченна (как и в большинстве коммерческих Unix). Но в **/usr/foss** вы найдете настоящий оазис – набор признанных свободных инструментов.

Также обратите внимание, что окружение по умолчанию ориентировано скорее на администраторов, чем на разработчиков и конечных пользователей. Никаких подсвеченных ls-команд, виртуальные терминалы не активированы, и с большинством утилит почти отсутствует обратная связь. Если вы опробовали одну из BSD-систем с диска *LXF77*, то уже поняли, что проблема решается парой дополнительных нажатий на клавиши. (Попробуйте **ls -F**, для получения более подробного вывода). Это лишь небольшая хитрость, поскольку Solaris не отвлекается на украшательства консоли, будучи ориентированным на управление мощными рабочими станциями и серверами.

В качестве настольной системы OpenSolaris (включая его вариант Belenix) ведёт себя подобно любому популярному Linux. Он надёжен, работает на порядочной скорости и поддерживает известные открытые приложения типа *Firefox* и *Gimp*. Однако исторически сложившаяся ориентация на машины Sun SPARC означает, что уровень



На освоение *DTrace* требуется время, но, к счастью, есть множество прекрасных скриптов, пригодных как основа ваших собственных сценариев профилирования.

поддержки оборудования PC, аналогичный Linux или FreeBSD, не обеспечивается – возможно, вам не удастся заставить работать все устройства. Поскольку Solaris разработан для рабочих станций и серверов, а не для настольных домашних систем, отсутствие поддержки видеокарты UltraMegaWhizzBang 4000 не является главным критерием выбора, но будем надеяться, что проект OpenSolaris выведет его на один уровень со своим открытым собратом.

Профилирование с DTrace

Даже учитывая быстрое вторжение Linux на рынок серверов, здесь всё ещё сильна конкуренция между коммерческими поставщиками Unix (Sun, HP и IBM), и исследования и разработки приводят к крупным обновлениям. Говоря о Sun, её самые последние разработки – это *DTrace* и Zones, и мы искренне аплодируем компании, сделавшей такие технологии доступными под открытой лицензией. Давайте взглянем на эти новомодные функции…

DTrace, система динамической трассировки, считалась одной из самых раскрученных функций в Solaris 10, выпущенном в январе 2005 года; и по праву. Устанавливая «щупы» (probe) *DTrace* в различных частях системы, администратор может определить узкие места производительности и убедиться, что всё работает как часы. Этими щупами динамически управляет ядро: оно устанавливает их в пространство исполняющихся процессов, и когда они срабатывают, можно запустить в ответ какие-то собственные действия.

Что делает *DTrace* столь мощной, так это отсутствие необходимости вносить изменения в систему – ни тебе отладочных режимов, ни перекомпиляции программ. Она разработана для использования на промышленных машинах для решения реальных проблем. Сложновато? Лучший способ во всём разобраться – посмотреть её в действии. Когда вы загрузитесь в Belenix, наберите и выполните следующий скрипт:

```
#!/usr/sbin/DTrace -s
syscall::write::entry
/execname == «bash»/
{
                printf(«Bash PID %d called
write.\n», pid);
}
```

DTrace использует собственный язык сценариев «D», представляющий собой смесь C и Sh. В приведённом примере мы сперва вызываем интерпретатор *DTrace* и велим ему установить щуп на системный вызов **write**. Затем в строке **execname** мы указываем, что хотим узнать, когда оболочка *Bash* выполнит этот системный вызов. Так что, запустив этот скрипт, вы получите информацию, когда *Bash* вызовет *write*. Это базовый пример, но его достаточно, чтобы увидеть, как всё это работает.

В сочетании с D, *DTrace* является безмерно гибким (и порой трудным!) инструментом, уже доказавшим свою полезность. Разработчики Belenix используют сценарии *DTrace* для повышения скорости загрузки системы, и если вы заглянете в **/opt/DTT/Bin**, то найдёте кучу полезных скриптов для мониторинга производительности, сетевой активности и т.д.

Виртуальные зоны

Зоны – это виртуальные системы Solaris, нечто среднее между *chroot* и *VMware*. Например, вместо того чтобы устанавливать отдельные машины для почтового, web- и FTP-серверов или запускать их всех на



Belenix включает рабочий стол KDE (и *Xfce*), так что вы будете в своей стихии – даже в совершенно новой среде Unix.

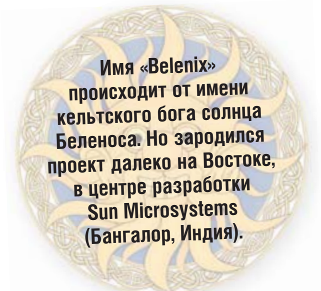
КАК Я МОГУ…

ВОЙТИ В СИСТЕМУ?
По умолчанию графический рабочий стол запускается автоматически, но если вам нужно ввести реквизиты, возьмите имя пользователя «root» и пароль «belenix».

НАЙТИ МОИ ДИСКИ?
Вы не найдете **/dev/hd*** или **/dev/sd*** – попробуйте заглянуть в **/dev/disks**, чтобы увидеть список узлов (node), представляющих ваши жёсткие диски. Числа соответствуют контроллеру диска, номеру диска и разделу или слайсу.

ЗАПУСТИТЬ X?
В отличие от большинства дистрибутивов Linux, команда *startx* в Belenix отсутствует; вместо неё нужно вводить **xinit** – запустится *X* с одним окном терминала. В этом терминале вы можете, набравшись храбрости, ввести **startxfce3**, **startkde** или даже **twm**…

ВЫКЛЮЧИТЬ СИСТЕМУ?
В командной строке наберите **shutdown -g0 -i0** (нули), затем **y** (это отличается от обычного синтаксиса Linux).



Имя «Belenix» происходит от имени кельтского бога солнца Беленоса. Но зародился проект далеко на Востоке, в центре разработки Sun Microsystems (Бангалор, Индия).

одном компьютере, что неминуемо влечёт проблемы безопасности, вы можете построить виртуальные машины для них на одном компьютере. Другими словами, с виду это похоже на отдельные установки Solaris, но внутри все они работают поверх одной реальной ОС – в так называемой глобальной зоне.

Виртуальные зоны независимы, и если одна «упадёт» или будет взломана, на остальные это не повлияет. Тем самым снижаются затраты на оборудование и время восстановления безопасной работы: если зона вашего виртуального web-сервера будет скомпрометирована, вы сможете за несколько секунд просто удалить её и скопировать поверх резервную зону. Существует множество других применений, но если вы пользовались другими открытыми инструментами виртуализации, например, *Xen* или *User-Mode Linux*, вам всё это будет знакомо.

Для создания зоны определяется каталог на жёстком диске, который будет использоваться как файловая система зоны, а затем указывается, должна ли зона запускаться автоматически. Можно также настроить для зоны сетевые интерфейсы, и когда она будет запущена, входить в неё как в обычную систему, используя **zlogin zone**. Одна из интеллектуальных функций зон – возможность наследования зоной части дерева каталогов основной системы, так что вам не потребуется копировать кучу файлов (и затем разгребать проблемы, если чего-то не хватит).

Основными инструментами настройки зон в Belenix являются *zonecfg* и *zoneadm* (сюда также входит скрипт *createzone*, пока не свободный от ошибок). Полный процесс создания зоны длинноват и в этой статье не уместится – мы просто попытаемся дать небольшое введение. Он, впрочем, не сложен, и вы найдёте руководство на нашем диске в файле **zonehowto.txt** (в секции **Distros/Belenix**). Чтобы разобраться с большинством функций зон, поищите более подробные руководства в Интернете: одни из лучших находятся на сайте Sun (**www.sun.com/bigadmin/content/zones**).

> «SUN МНОГО РАБОТАЕТ НАД ФОРМИРОВАНИЕМ СООБЩЕСТВА OPENSOLARIS.»

Солнце и тень

Итак, что ждёт OpenSolaris впереди? Показательно, что к середине февраля 2005 года более 11000 человек зарегистрировались на сайте проекта (из них в Sun работают только 1200), и было выполнено более 28000 загрузок исходного кода. Участие в реализации кода, впрочем, не столь велико: из 79 изменений в дерево кода, присланных сообществом, приняты только 34. Сильно беспокоиться не о чем, учитывая, что с момента открытия кода прошло меньше года, но нужно работать над улучшением ситуации. Будущие релизы основного продукта – Solaris – будут базироваться на OpenSolaris, так что чем больше участие, тем лучше.

В последние годы Sun много критиковали за явно двойственное отношение к Open Source; с одной стороны, она стремилась укрепить позиции Solaris в борьбе с Linux, но в то же время это был чрезвычайно полезный игрок сообщества в смысле приобретения и выпуска исходного кода *OpenOffice.org*. Теперь, когда большая часть Solaris открыта, есть надежда на появление ещё более тесных связей между Sun и миром свободного ПО. Будет ли свободный обмен кодом между Solaris и Linux, зависит от вопросов лицензирования; CDDL и GPL2 не совместимы, но ходят слухи, что Sun собирается перейти на другую лицензию. Поживём – увидим…

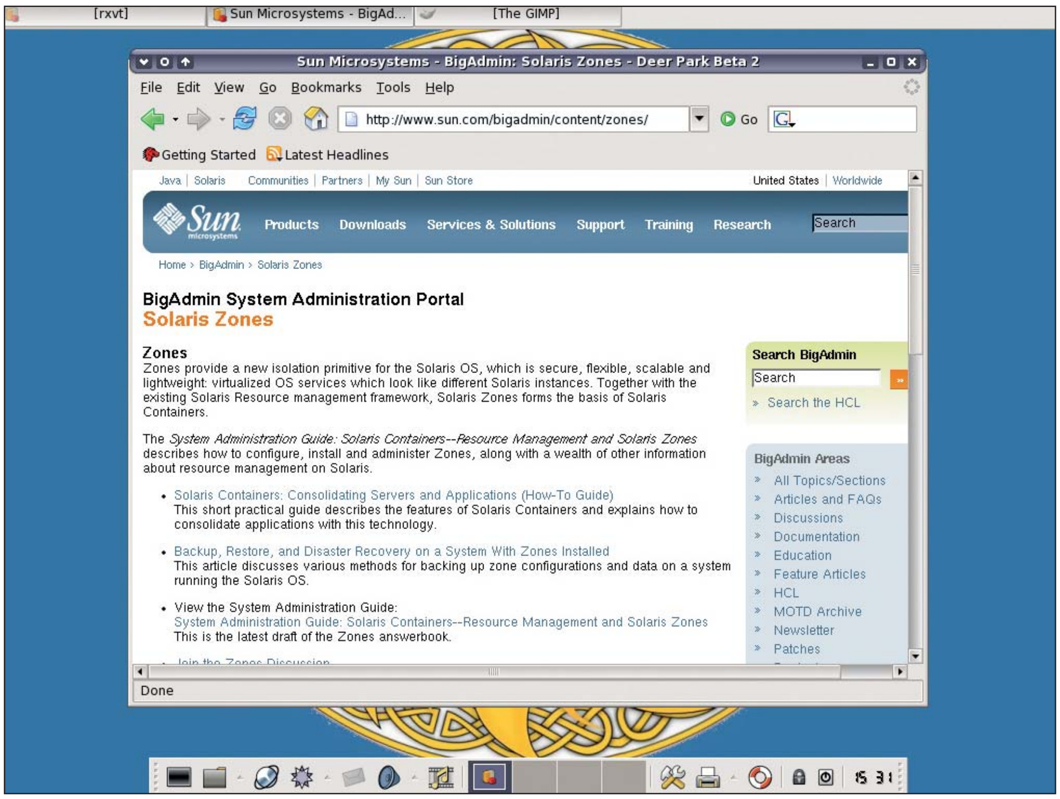
Но один код сам по себе не принесёт успеха, и Sun прилагает немало усилий, формируя для OpenSolaris сообщество, списки рассылки и доступную процедуру разработки. По всеобщему признанию, примерно то же происходит с *OpenOffice.org*, где большая часть пока что разрабатывается инженерами Sun. Вероятно, объяснением является чудовищная сложность кода *OpenOffice.org*, но мы надеемся, что Sun сможет создать вокруг OpenSolaris тот же уровень энтузиазма, что и у разработчиков Linux и BSD. Здесь заложен значительный потенциал – правильно использованный, он поможет всем: разработчикам свободного ПО, пользователям Solaris и всем, кто работает над каким-либо проектом операционной системы. LXF

РЕСУРСЫ

- **www.opensolaris.org**
Основной сайт разработки, с исходным кодом и списками рассылки
- **www.sun.com/software/solaris**
Официальная страница Solaris
- **www.gnusolaris.org**
Дистрибутив Nexenta OpenSolaris
- **http://schillix.berlios.de**
Дистрибутив Schillix OpenSolaris
- **www.blastwave.org**
Двоичные пакеты для Solaris
- **www.solaris-x86.org**
Информация о Solaris на PC x86



Nexenta стремится объединить лучшие черты GNU, Debian и OpenSolaris в одной ОС.



На сайте Sun много информации о зонах, реализующих виртуализацию в Solaris.

opensolaris

ВАШЕ МНЕНИЕ

Как вы думаете, зависит ли от OpenSolaris широкое принятие предприятиями Open Source? Черкните строчку на **letters@linuxformat.ru**.



Собираем Linux-киоск

Грэм Моррисон сочинил руководство по созданию интерактивного хранилища информации, подходящее для музея и Интернет-кафе, приёмной врача и выставки…

Киоски – не только торговля газетами и мороженым. Киоски есть и в мире высоких технологий, где вместо мороженого покупают информацию, а вместо газет – доступ в Интернет. Информационные киоски имеются во многих музеях мира – коснитесь экрана, и узнаете много нового об экспонате. Тот же киоск лежит в основе вашего Интернет-кафе: это обычный компьютер, только его внутренние капризы убраны с глаз долой, а снаружи остался удобный доступ в Интернет для забежавших попить кофейку.

Linux – превосходная система для строительства киоска: свободная, открытая и надёжная. Киоск даже дает способ заработать на вашем знании Linux. Многие музеи, библиотеки и предприятия могли бы извлечь реальную выгоду от разных видов киосков. Примечательно, что для этого не надо быть экспертом. На паре следующих страниц мы расскажем всё, что нужно знать для строительства собственной системы. Мы начнём с оборудования (процессоры, сенсорные экраны и пр.), затем обсудим тему защиты системы от случайного вторжения пользователя, а завершим полезными советами по созданию ваших собственных приложений для работы в киоске.

> «LINUX – ПРЕКРАСНАЯ ОС ДЛЯ СОЗДАНИЯ КИОСКА: ОТКРЫТАЯ И НАДЁЖНАЯ.»

В качестве примера приводится киоск на основе полноэкранного Интернет-браузера, привязанного к одному сайту – такие встречаются в музеях. Почти все понятия пригодятся и для любого киоска. Если идея информационного киоска вас не манит, подумайте об Интернет-кафе, они сейчас очень популярны. Всё, что вам нужно – обычная Linux-система, несколько старых процессоров и экранов и умение ограничивать доступ к определенным частям ОС. Мотивацией служат даже не деньги: разработка проекта – прекрасный шанс побольше узнать о Linux, а конечный результат может оказаться полезен местной библиотеке. Если вас когда-либо посещала идея о создании портала на сенсорном экране – самое время собрать в кучу дешевое оборудование и заставить его поработать!

Выбор оборудования

Долой быстродействие: киоск – это медленно, но верно.

Киоск никогда не выключается, и обычно расположен вдали от дома. Значит, не надейтесь на помощь эксперта – исправлять неполадки придется самим. Поэтому основные части системы – материнская плата, память и процессор – должны быть возможно более надёжными. По счастью, для большинства инсталляций не требуется сверхпроизводительности, и можно выбрать компоненты с солидной родословной, а необходимость грабить банк для покупки новейшего оборудования отпадает.

Охлаждение

Основное правило – чем лучше система охлаждается, тем стабильнее она работает. Следует выбирать процессор с невысокой рабочей частотой, а сочетание видеокарты и жесткого диска пусть обеспечивает скорее эффективность, чем быстродействие. Рискнём порекомендовать материнские платы EPIA от VIA. Температура процессора на них сравнительно низка, а низковольтные модели обходятся без вентилятора, обычно создающего неприятности.

Другое преимущество плат EPIA – у них выносной блок питания, так что второй по значимости источник нагревания убран из корпуса компьютера. Третий источник – видеокарта, производящая тепло в прямой зависимости от количества выполняемой работы. Вашим приложениям едва ли потребуется объёмная графика, поэтому видеокарта будет работать далеко не в полную силу.

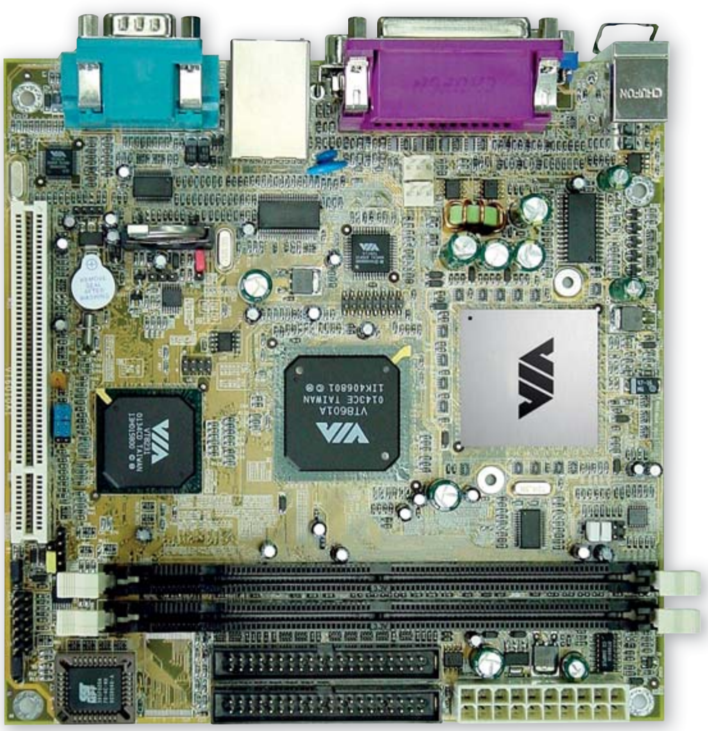
Исходя из этого, лучше раздобыть видеокарту без вентилятора, например, что-нибудь из линейки Nvidia FX5200 или старых Matrox. Чем меньше вентиляторов, тем меньше мороки с запчастями. И ещё, некоторые экраны могут работать только со старыми версиями *X Window*, имейте это в виду при выборе модели.

Другое слабое место можно укрепить, отдав предпочтение флэш-памяти перед винчестером. Это зависит от типа медиа, которое вы намерены использовать в приложениях вашего киоска. Если планируется видео и аудио, то емкости флэша может не хватить. А вот если ограничиться текстом и фотографиями, то на USB-флэш вместительностью в пару гигабайт небольшой Linux-дистрибутив и данные поместятся легко. Вообще-то и винчестеры не так уж плохи – они вместительны, недороги, и большинство из них в состоянии бесперебойно работать несколько лет.

>>



Чтобы сэкономить на хрупких деталях, поищите видеокарту Nvidia без вентилятора.



Если вы не требуете от вашей системы «сверхкрутизны» (да и зачем она вам), возьмите для киоска материнскую плату EPIA.

ГРАНИЦА НА ЗАМКЕ

Старая шутка системных администраторов утверждает, что для гарантированной безопасности вашей машины надо отключить её от сети. Зарыть сервер в бункер под 20 тонн бетона ещё надежнее! Ну, хоронить наш киоск мы не будем, а вот о включении его в сеть стоит поразмыслить. Конечно, сеть иногда неизбежна для поставки контента, а в случае Интернет-киоска вообще рассуждать не о чем. Вдобавок, если вы не живёте географически близко от вашего киоска, удаленный доступ – единственно возможный способ обслуживания и управления.

Это особенно касается малообслуживаемых и редко обновляемых систем, то есть большинства киосков. Как мы часто упоминали в этом журнале, SSH – единственное безопасное средство управления удалённым компьютером, и в прошлых номерах рассказывалось о безопасной работе с помощью разделяемых ключей (*см. статью «Блокируем хакеров», LXF 71, стр. 46*)

Если ваш киоск подключён к Интернету, важно настроить тип входящего и исходящего трафика для вашей машины. Для Интернет-доступа лучше всего взять прокси-сервер, открытый только для определенного контента. *Squid* – вероятно, самый популярный выбор, и он включен в состав большинства дистрибутивов по умолчанию. К тому же *Squid* достаточно просто настроить, особенно если вы пользуетесь инструментом *Webmin*. Если вы не трогали настройки web-доступа, то необходимо убедиться в том, что брандмауэр защищен от неправомочного использования с других сторон, например, от P2P-сетей или FTP.



Удалённый доступ через *VNC* с помощью SSH – лучший способ управлять киоском на расстоянии..

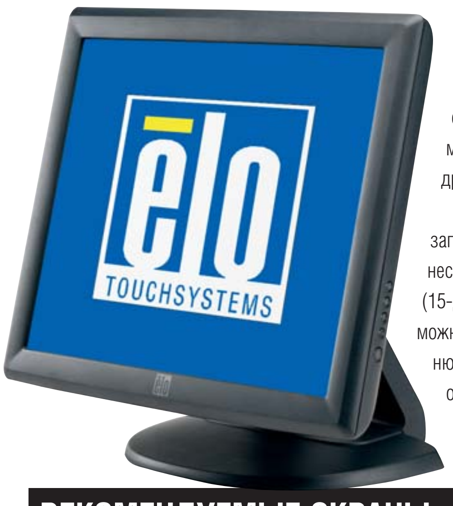
Экран должен быть приветливым

Об использовании тачскринов и трекболов под Linux.



Пользователям необходимы средства управления системой – успех или провал в этой области равносилен успеху или провалу вашего киоска. Если вы пытаетесь завлекать случайных прохожих, ваша система обязана быть дружелюбна и гостеприимна. Вот почему большинство подобных систем оснащаются сенсорными экранами (touchscreen). Прикоснувшись к такому экрану, человек тут же получает ответную реакцию, при этом глубокие компьютерные знания совсем не обязательны. Бремя тяжкой работы переложено на создателя пользовательского интерфейса, да и физическая природа сенсорного экрана рассматривается пользователями как дружественная форма технологии.

> «ПОЛЬЗОВАТЕЛИ ОБОЖАЮТ TOUCHSCREEN'Ы, ДЛЯ НИХ ЭТО ДРУЖЕЛЮБНАЯ ТЕХНОЛОГИЯ.»



Если вам не подходит сенсорный экран – возможны варианты, например, трекбол (trackball). Будучи неизмеримо дешевле, он к тому же работает под самыми обыкновенными «мышиными» драйверами.

Ещё недавно touchscreen'ы были запредельно дороги, но за последние несколько лет здорово подешевели (15-дюймовый стоит около $750), и можно усовершенствовать вашу нынешнюю машину за $300. Появился обширный выбор технологий; каждая по-своему хороша. Не думайте, что сенсорный экран всего лишь эмулирует работу с мышью: он не просто таскает курсор. Вообще, существуют четыре вида touchscreen'ов:

- **Чувствительные к нажатию** При касании экрана меняется сопротивление тонкого металлического слоя; изменение обнаруживается и используется для определения координат точки контакта. Этот метод не очень точен по сравнению с остальными, зато безразличен к контактному материалу – это может быть палец, карандаш, что угодно.
- **Электрочувствительные** Электроемкость панели изменяется внутренней ёмкостью «подключенного» человеческого тела. Поэтому реакция возникает только на нечто, проводящее ток, например, палец. Тот же механизм используется в координатных площадках (touchpad) ноутбуков, и реакция получается весьма точной.
- **Звукочувствительные** Ультразвуковые колебания играют ту же роль, что и электричество в двух предыдущих методах. Поверхность экрана покрыта сетью чувствительных к ультразвуку ячеек, и воздействие на них дает очень точный результат. Современные версии дают разрешение до 4096x4096.
- **Фоточувствительные** Экран покрыт сеткой инфракрасных излучателей и светочувствительных датчиков, и если луч света прерывается, то точку контакта можно рассчитать очень точно. Одно из преимуществ этой технологии состоит в том, что вы можете расположить группу излучателей вокруг монитора, а не покупать пакет целиком.

Конечно, говорить будет не о чем, если мы не сумеем заставить эти технологии работать под Linux. Основная забота – это совместимость драйверов и слаженность работы экрана и системы. Большинство современных устройств соединяются через USB, а старые чаще пользуются последовательным портом.

К счастью, большинство изготовителей экранов в той или иной мере поддерживают Linux. Поддержка обычно выражается в компилируемом Xinput-драйвере, который на старых версиях *X Window* нуждается в настройке. Файлы конфигурации драйверов часто ориентированы на *XFree86 3.3.6*, но большинство из них обойдется без проблем и под *XFree 4.0*, и под более свежими версиями *X.org*.

Без мышей

Когда модуль скомпилирован, остаётся отредактировать конфигурационный файл *X Window* для работы с экраном вместо мыши. Необходимо также добавить скомпилированный модуль в секцию «module». В конфигурационном файле *X Window* необходимо указать минимальные и максимальные X- и Y-координаты дисплея, они перечисляются в отдельной секции файла вновь созданного устройства.

Ниже приведён пример типичной конфигурационной секции сенсорного устройства, подключённого к последовательному порту. Предусмотрены поля шириной в 100 точек, недоступные из-за рамки монитора.

```
Section «Inputdevice»
Identifier «touchscreen1»
Driver «elographics»
Option «Device» «/dev/ttyS0»
Option «AlwaysCore»
Option «screenno» «0»
Option «MinX» «100»
Option «MaxX» «900»
Option «MinY» «100»
Option «MaxY» «900»
EndSection
```

Альтернатива покупке специализированного экрана – приобретение накладной панели на плоский экран. Необходимо убедиться, что размер панели соответствует вашему экрану и что она действительно вам подходит по размеру, но в остальном это – превосходный вариант, если у вас есть лишний ЖК-монитор.

РЕКОМЕНДУЕМЫЕ ЭКРАНЫ

ELO Touchscreens предлагает широкий выбор экранов – как на плоских мониторах, так и на мониторах с электронно-лучевыми трубками. Наряду с драйверами для Linux, ELO поставляет рудиментарную утилиту установки границ экрана, которая экономит время на редактировании файла настройки.
www.elotouch.com

Кроме самостоятельных устройств, Magic Touch производит накладные панели, вместе с Linux-драйверами для них, и их можно пристроить на любой стандартный ЖК-монитор из тех, что вы забросили на чердак. Это наиболее дешёвый вариант для того, у кого есть пара лишних экранов.
www.magictouch.com

Настройка ПО

Какой дистрибутив, программы и настройки выбрать для киоска?

Теперь, когда «железо» собрано и настроено, основная забота – операционная система. Требования к ней исчерпываются двумя словами: надёжность и безопасность. Конечно, они важны для любой системы, но машине, испытывающей постоянный стресс под нажимом многочисленных посетителей, эти качества жизненно необходимы.

Идеальный дистрибутив должен быть легким и хорошо протестированным. Этому определению соответствуют немногие, и мнения о наиболее подходящем, естественно, расходятся. Мы получили неплохой результат при работе в Ubuntu, это хороший дистрибутив с приличной поддержкой. Если нужна повышенная безопасность – поищите что-нибудь вроде Fedora Core 5 с SELinux или (для любителей *AppArmor*) SUSE 10.1. Для работы в Интернет-киоске может понадобиться web-сервер, установите его и убедитесь в работоспособности. Большинство дистрибутивов готово для работы с *Apache*.

Менеджер окон

После выбора дистрибутива встает вопрос: какой рабочий стол лучше? Ответ зависит го. Однако при работе с некоторыми видами графического интерфейса ограничений учётной записи может оказаться недостаточно, будь то просто расширенный браузер или более изощрённый рабочий стол.

Золотое правило гласит, что чем меньше вмешиваешься в работу системы, тем надёжнее она будет работать, поэтому лучший способ обезопасить графическую оболочку – не запускать её вообще. Многие дистрибутивы имеют вариант входа под названием «filesafe», что означает работу в *X Window*, но без оконной среды (Gnome или KDE). В Ubuntu, выбрав рабочий стол Filesafe, вы это самое и получите. Останется запустить *Firefox* или *Konqueror*.

Теперь о более развитых системах. Вы удивитесь, но при всей его тяжеловесности и избыточности лучшим выбором представляется KDE.

Раздача приоритетов

KDE может позволить пользователю как угодно много или как угодно мало, в зависимости от иерархии файлов конфигурации (в Gnome 2.14, кстати, для этого есть специальный инструмент под названием Pessulus, но говорить о нём всерьёз ещё



Без излишеств: выбрав *Filesafe* во входном окне Ubuntu, вы получите конфигурацию *X*, которую справедливо можно назвать минимальной.

> «НЕТ ПРОГРАММЫ – НЕТ ПРОБЛЕМЫ, ПОМЕНЬШЕ СТАВЬТЕ ИХ НА КИОСК.»

рано). Правила, установленные администратором, имеют абсолютный приоритет перед -пользовательскими настройками. Администратор может запретить изменять фон рабочего стола или настройки прокси-сервера.

Местоположение конфигурационных файлов KDE зависит от конкретного дистрибутива. Простейший путь найти их – набрать **kde-config --path config** из-под root. Пользователи SUSE получат в ответ нечто вроде:

```
/root/.kde/share/config/:/etc/opt/kde3/
share/config/:/opt/kde3/share/config/
```

Mandriva будет лаконичнее:

```
/root/.kde/share/config/:/etc/kde/
```

Вывод обоих дистрибутивов показывает приоритеты файлов настройки: пользовательская директория (в данном случае, root) располагается слева, а глобальное местонахождение файлов – справа. Если вы когда-либо сами редактировали файлы настройки KDE, то картина вам знакома. В вашей домашней директории находится папка **~/.kde/share/apps** с файлами данных для приложений KDE, и **~/.kde/share/config** с файлами настройки упомянутых приложений. Например, **kdesktoprc** содержит настройку фона рабочего стола.



Если вашим пользователям захочется чего-нибудь помимо браузера, есть смысл настроить что-то вроде KDE.

от вида вашего киоска. В большинстве случаев весь экран занят браузером, тогда лучше взять наилегчайший оконный менеджер с минимальным набором функций. Далее мы рассмотрим идеального кандидата – *Filesafe* из Ubuntu. А как быть, если посетитель Интернет-кафе захочет выйти за пределы браузера, обратиться к офисным приложениям или скинуть на флэш добытые сведения? Тут нужен рабочий стол, который надёжно и легко запирается.

Для безопасной работы первым делом надо создать пользовательскую учётную запись, под которой будут работать приложения киоска. Учётные записи, строго ограничивающие работу пользователей определёнными функциями – одно из главных преимуществ Linux в проектах вроде наше-

Большинство параметров в этих файлах понятны без объяснений. Вот пара примеров из **kdesktoprc**:

```
[Desktop0]
WallpaperMode=NoWallpaper
[ScreenSaver]
Enabled=false
```

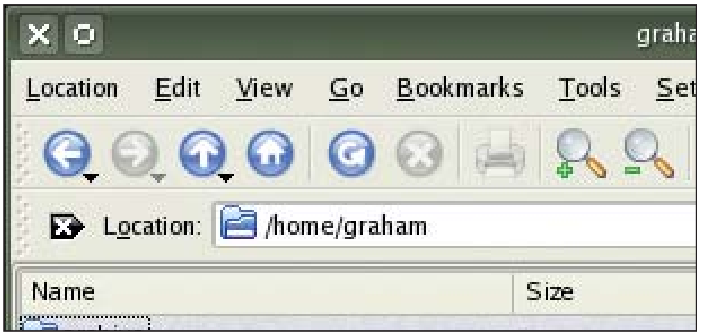
Если вам лень рыться в файлах конфигурации – можно войти в рабочий стол киоска через обычную учётную запись и создать необходимое вам рабочее окружение средствами KDE (вроде *KControl*). Новые настройки будут сохранены в директории **~/.kde** текущего пользователя (то есть вашей).

>>

<< Настройка пользовательского доступа

Неизменяемые параметры защитят систему от праздношатающихся пользователей.

Рассмотренная выше настройка может быть сделана штатными средствами KDE, да и почти любого другого рабочего стола. Для киоска важно, что параметры настройки, заданные в конфигурационном файле с высшим приоритетом, автоматически передаются «нижележащим» структурам. Например, системные директории всегда имеют приоритет перед домашними директориями пользователей. Но в пределах своей учётной записи пользователь волен менять настройки по своему усмотрению, что характерно для дистрибутивов с упором на графический режим работы. Нам остро необходима возможность уберечь параметры настройки от изменений, именно для этого и предназначен режим работы KDE «kiosk mode».



Action Restrictions избавил нас от меню закладок.

Ключ к «лишению свободы» – атрибут «неизменяемо» («immutable»): знак доллара, сопровождаемый строчной «i», заключённый в квадратные скобки **[$i]**. Если этот символ вставить после имени какого-либо параметра в файле настройки, имеющем высший приоритет по отношению к домашней директории пользователя, то параметр становится неизменяемым. Взяв за основу вышеупомянутый файл **kdesktoprc**, мы можем «запереть» фон рабочего стола таким образом:

```
[Desktop0]
WallpaperMode[$i]=NoWallpaper
[ScreenSaver]
Enabled[$i]=false
```

«Неизменными» можно назначать целые группы параметров. Группа – это список параметров под заголовком в квадратных скобках, например **[Desktop0]** или **[ScreenSaver]**. Чтобы «запереть» группу параметров, добавьте **[$i]** к концу строки: **[Desktop0] [$i]**.

Настроив отдельные приложения, можно обрезать некоторые опасные ветви KDE редактированием глобальных параметров, хранящихся в файле **kdeglobals**, в одной **config**-директории с уже отредактированными файлами. Параметры функциональности находятся под заголовком «KDE Action Restrictions» (Ограничение действий). По умолчанию, **kdeglobals** не содержит каких-либо ограничений.

Ограничения действий включают удаление и редактирование меню, замену значков рабочего стола и блокирование клавиатурных комбинаций. Ввод следующей строки в файл **kdeglobals** удалит меню закладок ***Konqueror***:

```
[KDE Action Restrictions][$i]
action/bookmarks=false
```

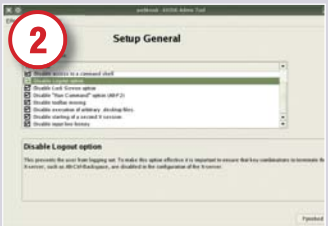
Полный список ограничений (а их, как нетрудно догадаться, немало) можно найти в документации по киоску KDE. Но KDE, как всегда, припас альтернативу: специальный инструмент *Kiosktool*. Это небольшое приложение по умолчанию включается в большинство дистрибутивов, содержащих KDE. Программу разработал Уолдо Бастьен (Waldo Bastien), и кстати, его же надо благодарить за многие удобства режима киоска KDE. *Kiosktool* – графическая оболочка для управления параметрами, которые мы рассматривали при изучении настроечных файлов. С помощью *Kiosktool* нетрудно найти нужный параметр и, что особенно важно, сгруппировать параметры в профили. Профили заметно упрощают переключение между различными уровнями функциональности или распространение конфигурации на группу машин (например, в Интернет-кафе).

НА ДИСКЕ

ИСПОЛЬЗОВАНИЕ KIOSKTOOL В KDE



Запустив *Kiosktool*, нажмите кнопку «Добавить профиль» (Add New Profile). Вам будет предложено дать профилю осмысленное название и ввести пароль суперпользователя. Чтобы копировать и редактировать файлы от имени суперпользователя, *Kiosktool* использует FISH (встроенный в KDE протокол SSH), отсюда его замедление по сравнению с обычным редактированием файлов, а также неожиданно возникающие непонятные прогресс-индикаторы – *Kiosktool* эффективно загружает и выгружает файлы. Всё это здорово для обслуживания удалённой системы, но не совсем удобно на локальной машине.



Щёлчок на вашем новом профиле откроет окно его настройки, там около12 значков, через которые легко добраться до нужной функции. Например, значок настройки меню KDE позволяет запретить изменения в меню и запуск задач от имени суперпользователя. Самое важное окно – «Общие» (General), здесь можно удалять закладки, контекстные меню, возможность выхода, команду Run (обычно доступную по F2) и историю команд. Без запрета доступа ко всем этим действиям киоску не жить.

Ограничение Firefox

Web-браузеры пресекают возможности пользователей киоска с солдатской прямотой. В *Konqueror'е*, например, достаточно просто отключить панели инструментов и меню (пометив отключение как «неизменное») в конфигурационном файле.

Большинство браузеров, включая *Konqueror* и *Firefox*, по клавише F11 переключаются в полноэкранный режим. Исчезают границы всех окон, а раз панели и меню недоступны, пользователю остаётся гулять по единственной Интернет-странице. То есть, пока он снова не нажмёт F11…

Лучший способ ограничить *Firefox*, превратив его в интерфейс киоска – воспользоваться расширением, которых, к счастью, существует несколько. Наиболее заметное из них – *R-kiosk*, его можно установить из окна расширений *Firefox*. Но будьте осторожны – произойдет переключение в полноэкранный режим, пропадут все панели инструментов и меню (в том числе контекстные) и отменятся клавиатурные комбинации. То есть, инструменты настройки отсутствуют, поэтому подготовьте все до запуска модуля (потренируйтесь на подходящей Интернет-странице). Для удаления модуля запустите *Firefox* в безопасном режиме (**firefox -safe-mode**) и удалите модуль *R-kiosk*.

Пять советов

Осталось самое приятное: сотворение интерфейса. Обратите внимание…



Попробуйте оживить разделы киоска. У нас изображение вдруг сползает с кружки.



Информационное наполнение киоска чаще всего опирается на web-дизайн. В этой статье не место обсуждать тонкости работы с HTML и другими Интернет-средствами, дадим только некоторые общие рекомендации. Интерфейс необходимо продумать до мелочей, вложив в его создание изрядную долю таланта, иначе все ваши старания окажутся пустой тратой времени. Дизайн – основа всего. При этом совершенно неважно, насколько грамотны ваши приложения JavaScript или PHP: только дизайн привлечет людей к вашей машине.

Для примера мы создали приложение-киоск, использующее обычный HTML и смысловые изображения. Приложение «оживляет» при помощи JavaScript некоторые области экрана, на которые можно щелкнуть для получения дополнительной информации. Это у нас экспонат компьютерной графики для музея. Прикосновениями к различным частям экрана можно вызвать информационную страницу и знакомиться с интересующим предметом в своем темпе.

Если вы хотите доставлять информацию в ваш киоск из Интернет, а не из локальной сети, то понадобится свой web-сервер (чаще всего пользуются *Apache*). Зная доменное имя «localhost», подключить приложение к серверу нетрудно. Почти каждый дистрибутив Linux содержит в себе установленный и запущенный *Apache*, а если нет – инсталляция требует всего нескольких щелчков мыши в панели управления дистрибутивом.

Взглянув на снимок экрана, обратите внимание на несколько деталей, которые помогут вам при дизайне киоска.

① БРОСКАЯ ВЫВЕСКА

Первое, что следует помнить – киоск должен приманивать людей, например, с другой стороны помещения. Значит, начальную страницу делаем простой и зазывной, ее задача – раздразнить интерес. Дизайн не должен быть перегружен, он должен притягивать, вызывать желание с ним поиграть. Задумывая дизайн киоска, неплохо почаще вспоминать о пирожном с выложенной ягодками надписью «Съешь меня» из «*Алисы в стране чудес*».

② БУДЬТЕ ПРОЩЕ

Не забывайте о свободном пространстве: минимум слов – вот девиз начальной страницы. С такого экрана неподготовленному, случайному человеку проще будет начать навигацию по информационным страницам киоска.

③ ИНТЕРАКТИВНОСТЬ

Пользователю необходимо видеть реакцию системы, особенно в отсутствие экранных инструкций. Можно создать интригующий интерфейс, в котором каждый последующий экран будет интереснее предыдущего. Анимация тоже способствует вовлечению пользователя в поиск очередной изюминки.

④ ЧЕТКАЯ ИЕРАРХИЯ

Сравните изображение на общем экране с видом экрана под цифрой ⑤, куда мы попадаем по щелчку. Идея киоска основана на взаимосвязи информационных страниц, а с любой из них необходимо предусмотреть возврат к началу. На экране ⑤ мы видим такое же изображение, только вид сверху; это создаёт ощущение другого угла зрения.

⑤ ТЕКСТ ДЛЯ СВЯЗИ

На этом уровне необходимо добавить вводный текст. Пользователи не любят долго читать, стоя столбом у машины, поэтому в каждую страницу мы встроили анимацию. В нашем случае для запуска анимации нужно нажать клавишу на клавиатуре.



Страницы, куда пользователь попадает по щелчку (например, такая) – идеальное место для размещения пояснительного текста.

НА ДИСКЕ

ЧТО ДАЛЬШЕ?

Самая сложная работа настает на этапе дизайна. Вариантов здесь так же много, как дистрибутивов Linux. Наше приложение создано на основе web-сервера, но с тем же успехом для этой цели можно использовать язык программирования. Разумеется, дорога вам проторена, и среди открытого ПО полно готовых программ для работы в Linux-киоске. Наиболее популярный для ограниченного использования (например, в библиотеках или Интернет-кафе) проект называется *OpenKiosk* (http://openkiosk.sourseforge.net).

Пишите, нам интересно будет узнать о ваших успехах в строительстве Linux-киоска. LXF

IBM WEBSPHERE COMMUNITY EDITION CONTEST 2006

Корпорация IBM и компания LinuxCenter.Ru представляют вашему вниманию конкурс IBM WebSphere Contest 2006! Это мероприятие нацелено на разработчиков, инструкторов, а также всех, кому интересны технологии Java и Open Source. Что нужно для участия? В первую очередь – установить в своей системе IBM WebSphere Application Server Community Edition (WAS CE). Если вы регулярно читаете наш журнал, то уже наверняка знакомы с этим открытым сертифицированным сервером J2EE-приложений, базирующемся на Apache Geronimo. Если же вы по каким-то причинам пропустили апрельский номер – не отчаивайтесь: всю необходимую для быстрого старта информацию можно найти во врезке. Дистрибутив IBM WebSphere Application Server Community Edition можно найти на прилагаемом к журналу диске: Сторона 2: Websphere CE/kick-start_ocd.iso. Перед установкой сервера данный ISO-образ необходимо предварительно записать на CD.

Основная задача конкурса – популяризация IBM WebSphere Application Server Community Edition/Apache Geronimo как открытого и эффективного средства для решения широкого круга различных задач, поэтому мы приветствуем не только готовые к использованию приложения, но и инструменты для интеграции WAS CE с другим свободным ПО (в первую очередь, Eclipse), проверки компонентов, шаблоны приложений WAS CE/Geronimo и так далее. Конечно, ваш творческий полет не ограничивается написанием кода – мы будем рады видеть методические материалы: серьезные статьи, планы учебных курсов, технические задания для учебного проекта и так далее.

Победители конкурса будут определены Экспертным советом, в который войдут представители ведущих софтверных компаний. Подведение итогов конкурса и торжественная церемония награждения победителей пройдет на выставке LinuxLand/SofTool'06 (Москва, ВВЦ, 26-29 сентября 2006 года). Победители конкурса в каждой из номинаций получат ценные призы, предоставленные организаторами конкурса:

– ноутбуки от IBM

– программное обеспечение, литература и атрибутика от LinuxCenter.Ru

– подписка на журнал Linux Format от редакции журнала.

Статьи и другие методические работы лауреатов конкурса будут опубликованы на сайте IBM developerWorks и на страницах генерального информационного спонсора – журнала Linux Format.

Впереди – целое лето, а как показывает опыт программы Google Summer of Code, за это время можно успеть многое. Желаем удачи в ваших начинаниях!

WAS CE – РУКОВОДСТВО К БЫСТРОМУ СТАРТУ

Для того, чтобы начать работу с текущей версией WAS CE (прежде всего, запустить этот сервер), необходимо:

>> Использовать одну из следующих операционных систем:

- Red Hat Enterprise Linux 3/4
- SUSE Linux Enterprise Server v9
- Windows 2003
- Windows XP SP2

Это те системы, на которых WAS CE прошел сертификацию на соответствие спецификациям J2EE 1.4, хотя сервер может работать и на других платформах.

>> Установить JDK 1.4.2_9 и старше, но не JDK 1.5.

Для запуска готовых приложений достаточно иметь Java Runtime Environemnt (JRE). На компакт-диске Java Kick-start находится JRE 1.4.2 от IBM, но можно использовать и JRE от Sun Microsystems.

Для разработки приложений необходимо наличие JDK 1.4.2 любого из вышеназванных производителей. Sun JDK (J2SE 1.4.2) можно бесплатно загрузить по адресу

http://java.sun.com/javase/downloads/index.html

После установки JDK имеет смысл добавить в переменную окружения **PATH** каталог **<jdk_install_dir>/bin**.

Хотя это и не относится непосредственно к WAS CE, многие Java-приложения используют переменную окружения JAVA_HOME. Ей можно присвоить значение в виде имени каталога установки JDK.

Теперь можно приступать к установке сервера с диска Java Kick-start.

Последнюю версию WAS CE можно также загрузить с сайта IBM по адресу:

http://www.ibm.com/developerworks/downloads/ws/wasce/?S_TACT=105AGX28&S_CMP=DLMAIN

Разработчики могут обращаться к документации как по WAS CE (входит в состав дистрибутива), так и по Geronimo. Документация по Geronimo доступна по адресу

http://geronimo.apache.org/.

Статьи на русском языке, объясняющие различные аспекты установки и использования Geronimo, можно найти также по адресу:

http://www-128.ibm.com/developerworks/ru/



УСЛОВИЯ КОНКУРСА

ОБЩИЕ ПОЛОЖЕНИЯ

1. Участвовать в конкурсе «IBM WebSphere Contest 2006» может любой человек, ознакомившийся с настоящими условиями и согласный с ними, за исключением сотрудников IBM, LinuxCenter.Ru, журнала Linux Format и членов их семей.
2. Учредителями конкурса выступают корпорация IBM и компания LinuxCenter.Ru, генеральным информационным спонсором – журнал Linux Format.
3. Конкурс проводится в период с 10 мая по 10 сентября 2006 года. Поведение итогов конкурса и церемония награждения победителей состоится в ходе выставки LinuxLand/Softool'2006 (Москва, ВВЦ, 26-29 сентября 2006 года). Итоги конкурса будут также опубликованы в ноябрьском номере журнала Linux Format.
4. Победители конкурса в каждой из номинация определяются Экспертной комиссией, состав которой утверждается Учредителями конкурса. Решение Экспертной комиссии является окончательным. Денежный эквивалент призов не выплачивается.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ РАБОТ

1. Присланные на конкурс работы должны попадать в одну из трех номинаций:

   • Методическое и информационное обеспечение.
   Это может быть большая, серьезная статья с детальным рассмотрением конкретных особенностей WAS CE версии 1.0 и выше; план учебного курса с учебными примерами и системой контроля усвояемости материала; грамотное техническое задание для учебного проекта: проработанная структура сайта на эту тему и т.п.

   Инструментарий
   Это могут быть расширения для Eclipse, облегчающие работу с WAS CE, утилиты для проверки компонентов, перекодировки дескрипторов для перехода на WAS CE с других J2EE-серверов, шаблоны (templates) для разработки приложений с использованием WAS CE.

   • Решения.
   Это готовые приложения, компоненты и любые законченные и готовые для использования фрагменты кода, устанавливаемые на серверах WAS CE/Geronimo.

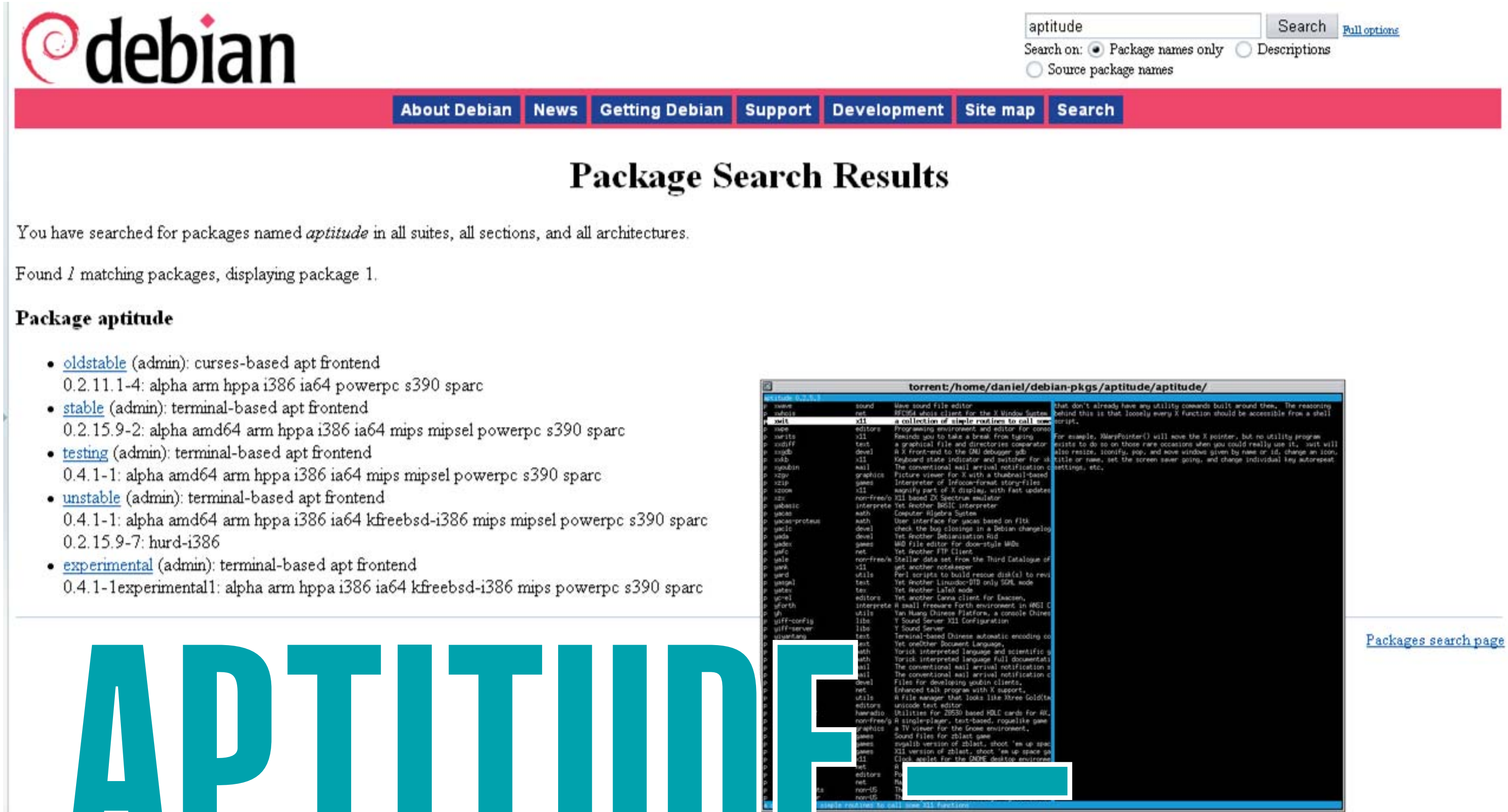
2. Имущественные авторские права на работы, представленные в номинации 1, должны принадлежать участнику конкурса. Это подразумевает, что статья или любая другая методическая разработка не может быть опубликована в печатных или сетевых изданиях (журналах, сборниках, сайтах и т.п.), ограничивающих право автора на размещение материала в других источниках. Код, содержащийся в статьях или методических разработках, должен быть представлен в форме, удобной для проверки его работоспособности и эффективности, то есть в виде полных исходных текстов и сценариев для сборки.
3. Работы, представленные в номинациях 2-3, должны распространяться на условиях, одобренных Open Source Initiative, то есть по какой-либо открытой лицензии. Распространение кода на тех же условиях, что и WAS CE/Geronimo – приветствуется.
4. Конкурсные работы следует высылать на адрес электронной почты: contest@linuxformat.ru Допускается (а в случае с работами, попадающими в номинации 2-3 – приветствуется) публикация конкурсных материалах на web-сайтах, не ограничивающих имущественные права автора (SourceForge.net, домашние страницы авторов и т.п.). В этом случае на адрес contest@linuxformat.ru следует выслать лишь ссылку на такой сайт с кратким описанием представляемой работы.

ПОРЯДОК КОНКУРСНОГО ОТБОРА

1. Представленные на конкурс работы будут оцениваться по следующим критериям:

   1) актуальность и востребованность (в том числе потенциальная – по мнению экспертов);
   2) качество решения и его оформления, включая документацию;
   3) степень универсальности решения.





APTITUDE — ПРЕВОСХОДНАЯ СТЕПЕНЬ АРТ

Подумали о морфологии? Напрасно – речь идет вовсе не о прилагательных. **Тихон Тарнавский** рассказывает о новом прогрессивном способе управления пакетами в *Debian*.

О менеджере пакетов *Debian* – *apt* – известно уже давно и многим, и рассказывалось о нём не раз; и это не странно, так как он определённо заслуживает внимания. Странно другое: что вниманием этим до сих пор обделена программа, призванная заменить пользователю две основные утилиты из набора apt и уже сейчас справляющийся со своими обязанностями много лучше них. «Куда уж лучше?» – спросите вы. Читайте дальше. Здесь я расскажу о почти универсальном инструменте управления пакетами – *aptitude* (**http://packages.Debian.org/aptitude**) – и его богатых возможностях. И думается, новое и полезное в этих возможностях могут найти как апологеты командной строки, даже уже попривыкшие ко всему богатству, предоставляемому стандартными утилитами apt, так и приверженцы графических оболочек в лице *synaptic*. Потому как ни *synaptic*, ни apt-tools не обладают такими удобными и полезными функциями, как *aptitudec* – о них-то я и хочу вам рассказать. Ну и конечно же, новые пользователи *Debian* и основанных на нём дистрибутивов тоже могут найти для себя много интересного.

Программа *aptitudec* может работать как из командной строки, так и в интерактивном режиме, с которого мы, пожалуй, и начнём. Чтобы перейти в интерактивный режим, нужно просто запустить программу без параметров.

Азбука aptitude

После старта *aptitudec* перед нами предстаёт его незамысловатый текстовый интерфейс следующего вида (см. *рис. 1*).

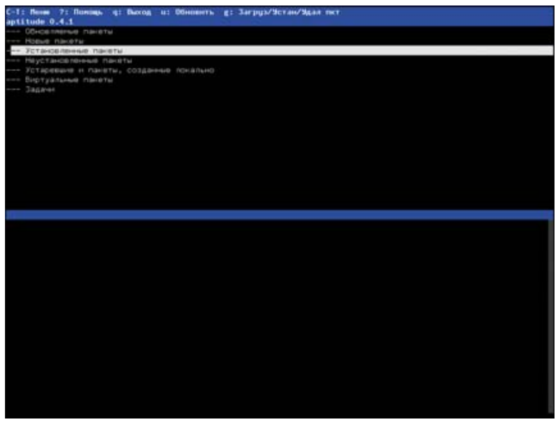


Рис. 1. Сразу после запуска.

В самой верхней строке экрана находится подсказка по основным клавишам, первая из которых, F10 или Ctrl-T, вызывает меню, появляющееся в той же верхней строке. Во второй строке в момент запуска написана только версия программы; в дальнейшем там будет появляться ещё сжатая информация об объёме, который будет дополнительно занят или, наоборот, освобождён при установке, удалении и обновлении выбранных вами пакетов, а также об объёме данных, которые необходимо будет скачать из Интернета. Остальная часть

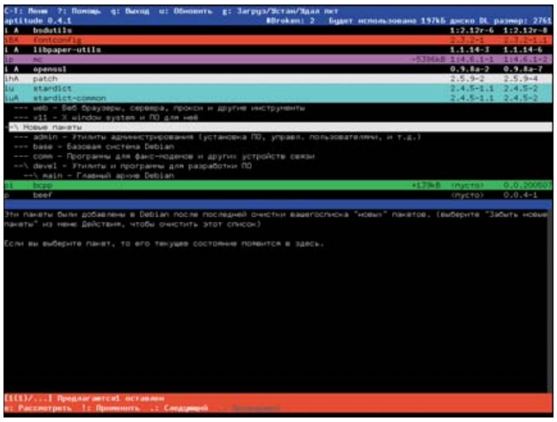


Рис. 2. Цветовая подсветка пакетов с разным статусом.

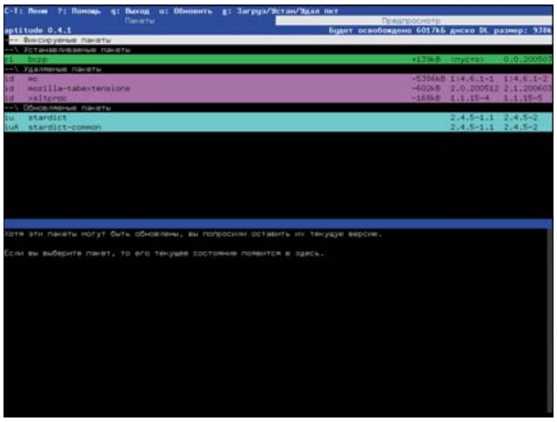


Рис. 3. Предпросмотр действий, запрошенных вами, перед их применением.

экрана по умолчанию разбита на две равные горизонтальные половины: в верхней отображается список пакетов, разделённый по категориям (каким именно – очень гибко настраивается, да и не столь важно в целях дальнейшего изложения); в нижней – описание того пакета или категории пакетов, которые выбраны в верхней половине в данный момент времени. В правой части списка пакетов напротив соответствующих имён расположены два столбца цифр. В первом из них отображается номер версии пакета, которая сейчас установлена в системе. Во втором – номер самой свежей версии, доступной во всех подключенных репозитариях; конечно, со сноской на Default-Release (см. «*Памятку об apt-репозитариях*»), если он задан.

Перемещаться вверх-вниз можно стрелками, клавишей Enter – сворачивать/разворачивать категории пакетов. Остальные управляющие клавиши в большинстве своём ничуть не более неожиданны: к примеру, для установки или обновления пакет отмечается клавишей **+**; для удаления – клавишей **–** ; для удаления вместе с конфигурационными файлами – «_», т.е. Shift+«**–**»; также можно зафиксировать пакет на текущей версии (т.е. запретить обновлять его при полуавтоматическом обновлении всего дистрибутива) с помощью клавиши **=**. Все эти отметки отображаются в интерфейсе программы своим цветом (*рис. 2*). Установленные в системе пакеты выделены жирным белым шрифтом. Пакеты, отмеченные для установки, выделяются зелёным фоном, для обновления – голубым, для удаления – фиолетовым. Белым фоном выделяются зафиксированные, а красным – сломанные (т.е. обладающие нарушенными зависимостями) пакеты. Текущее положение курсора отмечено инверсным выделением строки.

Как видите, перечисленными клавишами пакеты только помечаются для установки/удаления, никаких реальных действий по отдельному нажатию на каждую из них не происходит. После того, как вы выбрали необходимые пакеты для определённых действий, нажмите клавишу **g** (от «Go» – «поехали!») для того, чтобы выполнить их.

На самом деле, по первому нажатию ещё ничего не происходит – всего-навсего создается новый список всех пакетов, у которых должен измениться статус (*рис. 3*). Здесь, точно так же, как и раньше, можно скрывать/раскрывать категории пакетов; например, на приведенном рисунке свёрнута категория «фиксированные пакеты» – это те пакеты, которые оставлены не обновлёнными, хотя для них доступны более свежие версии. Одно из основных удобств в работе с программой – то, что все без исключения управляющие клавиши работают во всех без исключения видах списков пакетов (а их не так уж мало), причём не только на отдельных пакетах, но и на категориях. Например, в текущем списке вы можете установить курсор на строку «удаляемые пакеты» и одним нажатием на «_» удалить их вместе с конфигурационными файлами. Либо, если вы, к примеру, передумали обновлять все выбранные для обновления пакеты, нажать на строке «обновляемые пакеты» клавишу «**:**», которая отменяет обновление выбранных пакетов без их фиксации. После того, как вы убедитесь, что должны произойти именно те действия, которые вам нужны, вы можете ещё раз нажать **g** – теперь это уже приведёт к реальным операциям над пакетами (если вы запустили программу от имени обычного пользователя, то предварительно появится предложение ввести пароль root; думаю, не стоит акцентировать внимание на том, что *aptitudec* лучше запускать через sudo).

Сразу после повторного нажатия на **g** начнётся процесс загрузки всех запрошенных пакетов. Состояние загрузки будет отображаться несколькими индикаторами в низу экрана (*рис. 4*): одним общим и ещё по одному для текущего пакета на каждом из серверов, с которых производится загрузка.

Статусы пакетов, как вы могли заметить по двум предпоследним иллюстрациям, помечаются не только цветами. Обратите внимание на два крайних левых символа перед название каждого пакета на рис. 2 и 3: первый из них обозначает текущее состояние, второй – действие, которое затребовано в отношении данного пакета. Значения этих символов достаточно понятны и просты для запоминания: каждый их них является первой буквой названия (англоязычного, естественно) того действия или состояния, которое и обозначает (см. врезку «Легенда *aptitudec*»).

Вернёмся к экрану предпросмотра действий над пакетами. Как видно на *рис. 3*, между меню и строкой статуса появилась ещё одна строка. Как нетрудно догадаться по её содержимому, она представляет собой перечень открытых списков. Когда мы нажали **g** и перешли в режим предпросмотра, основной список пакетов никуда не делся, а остался открытым на заднем плане. Вы можете перемещаться между открытыми списками с помощью клавиш F6/F7; можете закрыть текущий список клавишей **q**. Например, если вы изменили что-то в списке >>

РОДОСЛОВНАЯ APTITUDE

Герой нашей статьи является, на сегодняшний день, «вершиной эволюции» среди программ управления пакетами в *Debian* и основанных на нём дистрибутивах, таких как *Ubuntu*, *Mepis* или *Knoppix*. Родоначальником этого эволюционного древа стал dpkg, функционально аналогичный широко известному RPM. Dpkg способен производить определённые действия с одним отдельно взятым пакетом. Так как работать с каждым пакетом в отдельности очень неудобно, то в качестве надстройки над dpkg появилось семейство программ *apt*, условно называемое apt-tools, утилиты которого работают уже не с единичными пакетами, а со всей базой. Эти утилиты набрали такую популярность, что превратились в стандарт де-факто в управлении пакетами не только на исторической родине – в *Debian*; многие из них были переработаны и для обработки баз rpm-пакетов, и их совокупность даже стала стандартным системным менеджером пакетов в некоторых rpm-дистрибутивах (например, Alt Linux). В самом *Debian* развитие средств управления пакетами на этом не остановилось и смогло породить нечто ещё более универсальное и автоматизированное, уже как надстройку над программами из состава apt-tools. Действительно, *aptitude* может всё, на что способна основная пара из набора *apt* – *apt-get* и *apt-cache*, плюс ещё немного, а большинство из общих с ними функций стали ещё более удобными для пользователя. Кроме того, *aptitude* может работать как в режиме командной строки, так и в качестве интерактивной оболочки с текстовым интерфейсом на базе *NCurses*. Все эти преимущества и привели к тому, что рекомендованным средством работы с пакетами в нынешней стабильной ветке *Debian* является уже не apt-tools, а именно *aptitude*.

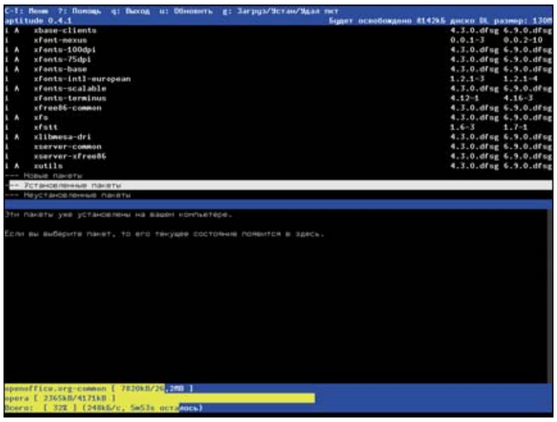


Рис. 4. Процесс загрузки пакетов с серверов с репозитариями.

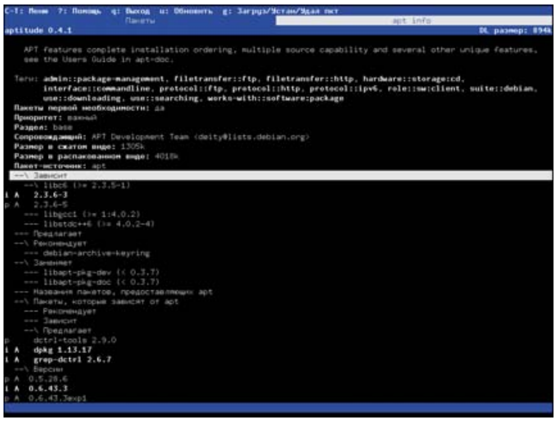


Рис. 5. Список свойств пакета.

<< предпросмотра, вы можете его закрыть (**q**) и затем вызвать снова, уже для новых действий (**g**). Это может оказаться удобным потому, что в списке предпросмотра отображаются ещё и такие пакеты, которые не выбраны для установки, но рекомендуются или предлагаются устанавливаемыми пакетами; после внесения изменений вы, возможно, захотите посмотреть на новые рекомендации. Естественно, перед выходом из предпросмотра не обязательно что-то менять – можно просто вернуться к общему списку «по собственному желанию».

Я уже упоминал о том, что списков отображения в *aptitudec* довольно много; а рассказал пока только о двух из них – общем списке пакетов и списке предпросмотра. Третьим основным списком является список свойств пакета. Попасть в него можно, нажав Enter на любом пакете (на рис. 5 показана часть такого списка для пакета apt). Этот список также содержит свои категории, которые точно так же можно скрывать и отображать. В их число входит список зависимостей и обратных зависимостей разного уровня, описание пакета и список доступных версий. Такие списки довольно наглядны, но в большинстве своём слишком громоздки для того, чтобы работать с ними целиком. Поэтому их части доступны как отдельные списки, вызываемые своими управляющими клавишами. Вот наиболее часто используемые из них: список версий, доступный по клавише **v** (versions); список зависимостей – клавиша **d** (dependencies); и список обратных зависимостей, т.е. тех пакетов, которые сами зависят от данного – клавиша **r** (reverse dependencies). Особенно полезно бывает использовать эти возможности в сочетании: например, списки зависимостей часто бывают разными у разных версий одного и того же пакета. К зависимостям конкретной версии можно попасть из списка версий, выбрав в нём нужную и нажав на ней **d**.

ПАМЯТКА ОБ APT-РЕПОЗИТАРИЯХ

Хотя это и не относится напрямую к *aptitude*, а касается и лежащего в его основе apt, напомним пару моментов касательно репозитариев deb-пакетов. Вероятно, Debian'щикам-новичкам эта информация окажется не лишней. Первый и основной файл настроек – это, конечно же, **/etc/apt/sources.list**. Вот простейший пример этого файла:

```
deb ftp://debian.org.ua/debian stable main contrib non-free
deb ftp://debian.org.ua/debian testing main contrib non-free
deb ftp://debian.org.ua/debian unstable main contrib non-free
deb ftp://debian.org.ua/debian experimental main contrib non-free
```

Второй момент, на который мне хочется обратить ваше внимание – это файл настроек **/etc/apt/apt.conf**. Конкретно, я хочу напомнить о такой опции, как *Default-Release*. Пример строки из файла apt.conf:

```
APT::Default-Release "testing";
```

Дело в том, что по умолчанию apt, равно как и *aptitude*, устанавливает для каждого пакета самую свежую версию, не глядя на то, в какой ветке она находится. Если же установлена упомянутая опция, apt и aptitude обращаться в первую очередь к указанной в ней ветке и, если нужный пакет в ней присутствует, выбирают именно его. Таким образом можно оставить всю систему в состоянии *testing*, даже если вы хотите установить один или несколько пакетов из *unstable* или даже *experimental*.

Далее углубляться в возможности тонкой настройки с помощью этих двух файлов я не буду. Наиболее полную информацию о них вы можете найти в соответствующих man-страницах: **man sources.list** и **man apt.conf**.

Ад зависимостей

К чему я так много говорю об этих списках? Сейчас мы обсудим ещё одну важную тему. Важную в том смысле, что это одна из ключевых проблем всех существующих систем управления пакетами, и *aptitudec* даёт практически неограниченные возможности для ручного решения этой задачи. Речь о «сломанных» зависимостях. Например, вы выбираете какой-нибудь один пакет для обновления, и его новая версия зависит от новой версии некой библиотеки; в то же время среди оставшихся необновлённых попался пакет, зависящий от более старой версии той же библиотеки (временами бывает, что в зависимостях версии прописаны строго, а не через >=) – всё, как минимум один из этих пакетов сломался. Что делать в таком случае? Всё довольно просто. По клавише **b** находим сломанный (broken) пакет, заходим в его список версий. Действительно ли нам нужна именно сломанная версия? Если нет, выбираем другую (так же, как и в основном списке: **+**) – возможно, на этом проблема будет решена. Если нужна именно сломанная версия, посмотрите на нижнюю половинку экрана, там может быть уже приведена причина поломки (вместо описания пакета); если нет, то переключение между описанием и служебной информацией осуществляется клавишей **i** (info). Описание причины даёт вам понять, в какую сторону двигаться: в направлении прямых или обратных зависимостей (**d** или **r** соответственно).

В списках зависимостей также работает поиск сломанных пакетов (**b**). Если вам все ещё не ясны пути решения, можно нажать Enter на подсвеченной красным строке, чтобы открыть полное описание соответственного пакета, а оттуда, прямо с заголовка – перейти в список доступных версий (**v**). Продолжайте, пока проблема не будет решена. После этого клавишей **q** закройте всю последовательность открытых списков и переходите к следующей поломке, либо, если таковых не осталось, приступайте к выполнению (**g**).

Ручное исправление поломок может помочь всегда – нужно только понимать, как именно их устранять в каждом конкретном случае, но это дело привычки. Кроме того, в версии *aptitudec 0.4.x* появилась новая возможность: автоматическое разрешение зависимостей. Как вы можете видеть на рис. 2, при наличии сломанных пакетов в самом низу экрана появляется рекомендация рассмотреть предложенные программой варианты. Нажав, как подсказывает *aptitudec*, клавишу **e**, мы попадаем в список пакетов, которые предлагается: сохранить на текущей версии, обновить, установить, удалить. Если предложенное решение вас не устроит, клавишей «**.**» можно перейти к следующему, а клавишей «**,**» – вернуться к предыдущему. Когда подходящее вам решение будет найдено, вы можете его применить, нажав «**!**». Оптимальное решение рано или поздно обязательно будет найдено, но нередко бывает проще разрешить зависимости вручную, чем пересмотреть с полсотни предлагаемых вариантов. Именно поэтому я и уделил столько внимания ручному управлению – вам может пригодиться и то, и другое.

ЛЕГЕНДА APTITUDE

Если вы подумали о принцессах и рыцарях, то ошиблись – легендой зачастую называют и банальный список условных обозначений. Ниже перечислены символы, обозначающие состояний пакетов и действия над ними.

i	installed (установлен)	install (установить)
d	–	delete (удалить без файлов настройки)
p	purged (не установлен)	purge (удалить полностью)
u	–	upgrade (обновить)
c	config (удалён без файлов настройки)	–
B	broken (сломаны зависимости)	broke (будут сломаны зависимости)

APTITUDE ВМЕСТО APT

Если вы уже привыкли к apt, то на первых порах можете использовать эту табличку для поиска соответствий между ключами командной строки в apt и aptitude.

aptitude	apt
aptitude install / remove	apt-get install / remove
aptitude purge	apt-get remove --purge
aptitude reinstall	apt-get install --reinstall
aptitude update / upgrade / dist-upgrade	apt-get update / upgrade / dist-upgrade
aptitude search	apt-cache search
aptitude show	apt-cache show
aptitude clean / autoclean	apt-get clean / autoclean
aptitude markauto / unmarkauto	нет аналогов

Ты где, пакетик?!

От приемов поиска сломанных пакетов мы плавно переходим к ещё одной очень объёмной теме: поисковым возможностям. Здесь я уже не буду вдаваться в подробности, а только перечислю некоторые функции. Поиск (так уж повелось) вызывается клавишей «**/**» и является инкрементным, т.е. выбор первого пакета, соответствующего критериям поиска, происходит прямо в процессе ввода. По умолчанию поиск ведётся по именам пакетов, но, используя управляющие символы, можно искать, к примеру, по описаниям или по состоянию пакета. Например, упомянутая выше клавиша **b**, поиск сломанных пакетов, – это всего лишь сокращение к «**/~b**», где «**~**» как раз и обозначает поиск по состоянию пакета. Аналогично можно искать, например, пакеты, от которых остались только конфигурационные файлы: «**/~**c». Когда вы полностью набрали искомую строку, вы можете закрыть окно поиска нажатием Enter и далее искать следующие вхождения этой же строки столь же традиционной клавишей **n**.

Ещё одно свойство *aptitudec* – оно же кардинальное отличие от apt-tools – никак нельзя обойти вниманием. В *aptitudec* пакеты имеют дополнительное свойство, говорящее о том, запросили ли вы этот пакет к установке вручную, или он установился автоматически для удовлетворения зависимостей. Сделано это для реализации одной очень удобной функции: автоматического удаления автоматически установленных пакетов, от которых уже ничего не зависит (лихо завернуто!). Это практически избавляет вас от необходимости ручных чисток. Пакет остаётся установленным у вас в системе только в двух случаях: либо вы сами решили, что он вам нужен, либо от него зависит что-то вам нужное; это сводит практически к нулю вероятность что-то случайно забыть в установленном состоянии. Кроме того, существует возможность самостоятельно пометить любой пакет как установленный вручную либо автоматически. Делается это соответственно клавишами **m** (от «manual») и **M** (**Shift+m**, вероятно, по аналогии с предыдущей). Эти клавиши, как и любые другие, действуют не только на отдельных пакетах, но и на категориях в любом списке. Например, вы можете нажать **M** на категории «**libs**», разом сделав все библиотеки установленными автоматически; т.е. те из них, от которых никто не зависит, будут удалены. Таким образом одним нажатием клавиши вы добьётесь действия, аналогичного умолчательному поведению отдельной утилиты управления пакетами – *deborphan*.

Это, пожалуй, практически всё, что я хотел сказать о работе с *aptitudec*. То есть, говорить, конечно, можно ещё очень и очень долго – возможностей он предоставляет несметное количество, и многие из них заслуживают внимания. Но всем сказанным почти исчерпалось то, чего не рассказать было просто нельзя. Под конец, для полноты картины, кратко затронем работу из командной строки.

CLI – великий и ужасный

В командной строке *aptitudec* предоставляет практически те же возможности, что и apt-get с apt-cache; чтобы не быть многословным, просто приведу список соответствий их параметров (см. врезку). Деление пакетов на «автоматические» и «установленные вручную» сохраняется и в режиме командной строки. Есть и ещё некоторые, менее существенные отличия от apt-tools, о которых вы можете прочитать в man *aptitudec*.

И в завершение мне остаётся только присоединиться к рекомендациям создателей дистрибутива: пользуйтесь *aptitudec*; думаю, вам понравится. **LXF**

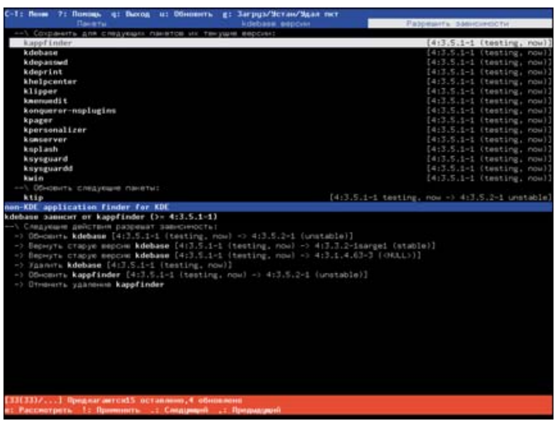


Рис. 6. Автоматическое восстановление нарушенных зависимостей.



Шрифты «как в Microsoft Windows»

В Сети можно найти множество рецептов облагораживания шрифтового хозяйства Linux. **Владимир Попов** предлагает целостный взгляд на эту проблему.

Эта статья – ещё одна попытка помочь сделать рабочий стол *nix-систем более привлекательным. Нынешние KDE или Gnome предлагают десятки стилей оформления окон, эффекты анимирования, плавного исчезновения, полупрозрачности и теней, а нарекания на внешний вид *X Window* по-прежнему нередки. В большой степени, как мне кажется, это определяется качеством рендеринга (прорисовки) шрифтов. Так ли плоха сама система рендеринга? Нормально ли, что одни и те же параметры можно задавать на уровне *X Window*, менеджера входа в систему, оконного менеджера и, наконец, отдельного приложения? И как с этим бороться?

Не буду пытаться советовать, как именно «сделать красиво». Постараюсь лишь обратить внимание на некоторые технические аспекты рендеринга – в надежде на то, что кому-то это поможет создать рабочий стол на базе KDE или Gnome, о котором можно будет сказать: «Да, он действительно как в MS Windows или лучше».

Давным-давно...

Лет пятнадцать назад, беседуя с одной американской знакомой, впервые доставившей в Москву персональные компьютеры с процессором DEC Alpha, я услышал такую фразу: «Хочу привезти компьютеры с UNIX, чтобы вы увидели настоящую графику». О достоинствах UNIX в качестве серверной платформы я на тот момент знал достаточно, а вот о том, что её сильная сторона – графика, услышал впервые. После терминалов IBM 360 и PDP-11, после DOS и Norton Commander, графическая оболочка Windows NT, запущенная на этих самым Alpha'х, казалась едва ли не верхом совершенства. Признаться, я усомнился в правоте своей знакомой, но, оказалось, напрасно.

В те далёкие времена АРМ'ы (автоматизированные рабочие места), сделанные на UNIX (будь то рабочее место раскройщика кожи или тренажёр космонавта), действительно выглядели более презентабельно, чем MS Windows, делавшая тогда лишь робкие первые шаги. Шли годы, персональные компьютеры теснили мини-ЭВМ и мэйнфреймы, как млекопитающие динозавров, а вслед за ними и MS Windows становилась всё популярнее, тогда как о UNIX применительно к IBM PC сказать было нечего: его просто не было. Положение изменилось с появлением Linux. Идеи и наработки, накопленные за годы существования UNIX, стали постепенно находить себе место и на персональных компьютерах.

Оставим в стороне принципиальные различия между миром Open Source и проприетарным ПО. Сосредоточимся на графической оболочке *nix-ов. Как известно, все они используют открытую реализацию *X Window System*. По определению *X.Org Foundation*, *X Window* – это «прозрачная» сетевая система оконного типа, функционирующая на разнообразных вычислительных системах и столь же разнообразном графическом оборудовании. То есть: не только оболочка, не только для Linux и не только на IBM PC.

И это действительно так. Впечатляющие по своей экономичности клиент-серверные системы, в которых *OpenOffice*, *Mozilla* и Gimp успешно работают на станциях с процессором Pentium-166 и 64 МБ RAM – отличная тому иллюстрация. Серверные возможности *X Window* впечатляют, и трудно поверить, что в рамках того же проекта рендеринг шрифтов решён так безобразно, как можно предположить, выслушивая уничтожающие отзывы некоторых пользователей, сравнивающих рабочий стол Linux с Windows XP (с чем же ещё сравнивать?). С одной стороны, я беру на себя смелость защищать разработчиков *X Window*, а с другой – признаю, что для недовольства «простого» пользователя имеется подчас достаточно оснований. Ниже следует перечисле-

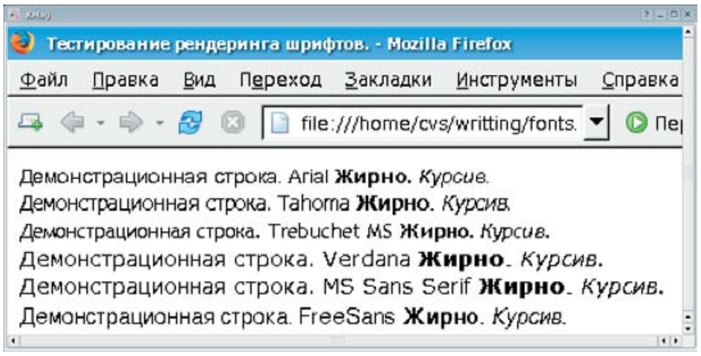


Рис. 1. Freetype собран с выключенным BYTECODE_INTERPRETER.

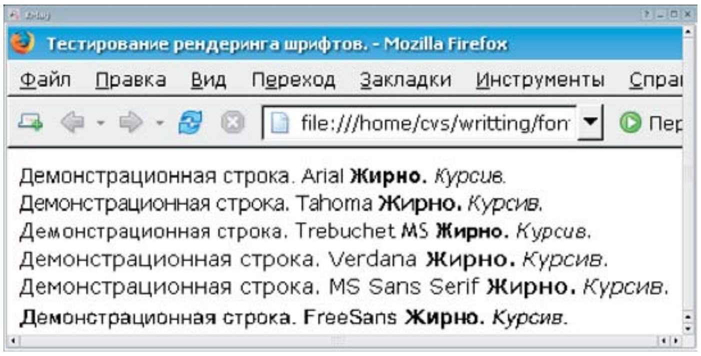


Рис. 2. А теперь – с включенным BYTECODE_INTERPRETER.

ние этих оснований с последующим предложением способа исправления ситуации.

libfreetype

Вообще-то не совсем справедливо, когда от *X Window* требуют, чтобы было не просто хорошо, а именно «как в XP». Реальность, однако, именно такова. И формулировка «как в MS Windows, даже лучше» (Like MS Windows, even better) достаточно точно отражает и характер, и уровень требований.

Прежде всего, это значит, что речь идёт о TrueType-шрифтах. Вывод растровых шрифтов давно уже не является проблемой, а использование свободных масштабируемых шрифтов, традиционно входящих в состав *X Window*, как бы совершенна ни была их прорисовка, оставляет пользователю, привычному к виду Windows, основания быть недовольным. Сила ли привычки определяет его отношение, или шрифты из состава *X Window* действительно так уступают **Arial/Tahoma/Verdana/Trebuchet** – не суть важно. Не будем также касаться правомерности использования TTF-шрифтов из состава Microsoft Windows. Зададимся иным вопросом: можно ли в *X Window* видеть их такими же, как в XP? Если «в принципе – да» , то почему буковки меньше и вообще как-то «аляповато»? Смотрим на рис. 1.

Первым и самым главным (в случае его наличия) недостатком является компиляция библиотеки *libfreetype* без опции **TT_CONFIG_OPTION_BYTECODE_INTERPRETER**. Существуют патентные ограничения на использования этого самого **BYTECODE_INTERPRETER**: патент принадлежит Apple, а действует ли он там, где может оказаться *libfreetype*, разработчики, разумеется, не знают. Так что и в версии **freetype-2.1.10** (последней стабильной на начало 2006 года) в файле **include/config/ftoption.h** опция **TT_CONFIG_OPTION_BYTECODE_INTERPRETER** не определена. Для незнакомых с синтаксисом include-файлов укажем, что файл **ftoption.h** должен содержать строку:

```
#define TT_CONFIG_OPTION_BYTECODE_INTERPRETER
```

Исходные тексты библиотеки можно взять на **freetype.sourceforge.net**. Желаемое местонахождение библиотек нужно указать в команде конфигурации, например:

```
./configure --prefix=/usr
```

если файлы должны оказаться в каталоге **/usr/lib**.

Компиляция запускается посредством make, инсталляция – **make install**.

Инсталляцию, в принципе, можно и не делать. Достаточно после компиляции переписать файлы **libfreetype.a**, **libfreetype.la** и **libfreetype.so.6.3.8** (для последнего файла в вашем случае версия, разумеется, может быть иной) в нужный каталог. Для «двоичного» дистрибутива такое решение даже предпочтительней: базу данных установленных пакетов редактировать не придётся [оптимальным с точки зрения «двоичного» дистрибутива выходом будет, конечно, пересборка пакета, но кто ж станет этим заниматься, – прим. ред.]. Если компилировалась версия libfreetype, отличная от ранее установленной в вашей системе, то можно переписать каталог **include** (в **/usr/include**) и конфигурационный файл **freetype2.pc**. «Можно», а не «нужно», поскольку если вы «простой пользователь», то **include** и **freetype2.pc** вам, скорее всего не понадобятся. Если же «прелести самосборки» вам не безразличны, то вы и сами все знаете – включая то, что после установки новых библиотек рекомендуется запускать **ldconfig**.

DPI

Когда с *libfreetype* будет всё в порядке, а рабочий стол перестанет вызывать приступы острой жалости, можно перейти к следующему шагу и попытаться ответить на вопрос: почему один и тот же шрифт одного и того же кегля выглядит имеющим разный размер под XP и *X Window*? См. рис. 2.

Дело в том, что ключевым параметром при создании векторных изображений является так называемый DPI – «dot-per-inch», который говорит о том, сколько точек (пикселей) помещается на экране в отрезке длиной дюйм. Казалось бы: разрешение известно, физический размер экрана – тоже, какие трудности? Тем более, что все современные мониторы имеют так называемый DDC (Display Data Channel) и по запросу выдают блок информации, в котором, среди прочего, должны присутствовать данные о физических размерах экрана. К сожалению, не-DDC мониторы пока не редкость и, самое главное, нет уверенности в том, что значение dpi, рассчитанное таким образом, обеспечит качественный рендеринг масштабируемых шрифтов.

Актуальное значение dpi всегда можно узнать, набрав в окне терминала:

```
xdpyinfo | grep resolution
```

Не так давно считалось, что достаточно двух значений dpi: 75 и 100. 75 – для вывода мелким шрифтом и 100 – крупным. Об этом до сих пор напоминают названия каталогов: **/usr/X11R6/lib/fonts/75dpi** и **/usr/X11R6/lib/fonts/100dpi**.

В то же время пользователи MS Windows знают, что им по умолчанию предлагается три размера шрифтов: малый, нормальный и большой (масштаб 72, 96 и 120 dpi, соответственно). Истоки различий значений по умолчанию я объяснить не могу. Не исключено, что они связаны с алгоритмами растеризации совершенно определённых фонтов. Во всяком случае, неоднократно отмечалась связь качества рендеринга TTF-шрифтов с кратностью задаваемого dpi шести и даже 12-ти. Так или иначе, но произвольные значения dpi, которые MS Windows позволяет устанавливать, воспользовавшись масштабом «Custom», достаточно часто приводят к заметному ухудшению отображения. Отметим пока эти три числа и вернёмся к *X Window*.

Воспользовавшись вышеупомянутой утилитой, можно узнать, каким *X Window* полагает размер экрана:

```
xdpyinfo | grep dimension
```

Выведенное в ответ разрешение экрана не удивляет (сами ведь задавали), размер экрана в миллиметрах получен, очевидно, от монитора, но вот насколько расчитанный dpi соответствует оптимальному рендерингу?

Поскольку в документации *X Window* рекомендаций по этому поводу найти не удалось (да и какой линуксоид не предпочтёт собственные эксперименты чтению документации?), впору перейти к описанию способов задания dpi. Итак:

- Начнём с **/etc/X11/xorg.conf**. В секции “Monitor” строкой

```
DisplaySize www hhh
```

можно задать те самые размеры экрана

>>

ТАК ЧТО ТАМ ЗА TTF?

Многообразие существующих ныне TTF-шрифтов не поддаётся описанию. Если, однако, исключить сугубо «декоративные», да ещё не содержащие символов кириллицы, то это число существенно уменьшится. При сравнении же с Windows и вовсе подразумевается вполне определённый набор – так называемый mscorefonts, куда помимо Arial, Courier New и Times New Roman от Monotype входят и наиболее известные шрифты MicroSoft: Verdana, Trebuchet и т.д. В свое время Microsoft распространяла эти шрифты как настоящее Freeware прямо с сайта **www.microsoft.com**. Сейчас фирма отказалась от первоначальных планов, но запретить распространять mscorefonts третьим лицам уже нельзя (Freeware все-таки), так что все необходимое можно найти на **http://corefonts.sourceforge.net**. Существующая лицензия не запрещает использование шрифтов в некоммерческих целях. Исключением является Tahoma (по совместительству – основной шрифт пользовательского интерфейса в Windows XP). Тут придётся считаться с ограничениями Internet Explorer EULA, которые, помимо прочего, предполагают, что у вас есть лицензия на Windows. Лицензия Freeware запрещает модифицировать дистрибутивные файлы, поэтому к пяти мегабайтам шрифтов придется прибавить ещё и утилиту cabextract – Microsoft не жалует общепринятые средства сжатия.

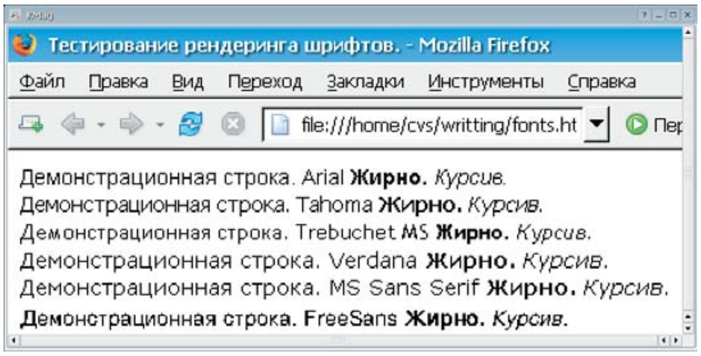


Рис. 3. На этом рисунке значение DPI равняется 96.

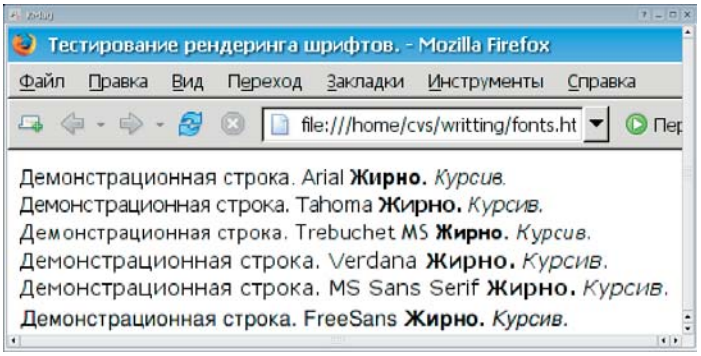


Рис. 4. Антиалиасинг может принести не только пользу

<<

в миллиметрах, которые **xdpyinfo** выдаст потом как **dimension**. Соответствующий dpi будет получен в результате простейших арифметических действий;

- Если *X Window* запускается из консольного режима, dpi можно задать в явном виде:

```
startx -- -dpi NN
```

или отредактировать сам скрипт **/usr/X11R6/bin/startx**, позаботившись о наличии строки:

```
defaultserverargs="-dpi NN"
```

- Если система запускает *X Window* при загрузке, то регистрация пользователя (и, опционно, выбор оконного менеджера) выполняется каким-нибудь менеджером входа в систему, многие из которых могут задавать dpi для открываемой сессии. Для kdm в конфигурационном файле **/opt/kde/share/config/kdm/kdmrc** в секции **[X-*-Core]** можно ввести строку:

```
ServerArgsLocal=-dpi NN
```

При использовании gdm строка запуска X-сервера в файле **/etc/X11/gdm/Xservers** должна выглядеть так:

```
:0 local /usr/X11R6/bin/X -dpi NN
```

- Системы, применяющие **fontconfig** (то есть, к настоящему моменту, практически все), часто используют конфигурационный файл **/etc/X11/Xresources**. В этот файл достаточно включить строку:

```
Xft.dpi: NN
```

- В некоторых дистрибутивах, производных от Debian, в файле **/etc/init.d/xsession** нужна строка:

```
DPI="-dpi NN"
```

Во всех перечисленных случаях NN – значение dpi. Что же касается того, каким должно быть это значение, то единого мнения нет. Никто не мешает поэкспериментировать, но если имеет место дефицит времени и желание воспользоваться «советами старых мастеров», то попробуйте dpi=75 или 100 для шрифтов из состава *X Window*, и наоборот – при «равнении» на MS Windows придерживайтесь ею же рекомендуемых значений (рис. 3).

Xft

Из двух систем, занимающихся визуализацией шрифтов в *X Window*, нас, безусловно, интересует *Xft*, поскольку речь идёт исключительно о масштабируемых шрифтах. Незатейливое **Xft-config --libs** сообщает, что в состав подсистемы, кроме X-библиотек ***Xft***, *X11* и *Xrender*, входит уже известная нам библиотека *freetype* и неизвестная пока *fontconfig*.

Дело в том, что *Xft* не имеет собственных средств конфигурации для задания характеристик шрифтов и параметров растеризации. Всем этим занимается библиотека *fontconfig*, а интерфейсом к ней являются файлы **/etc/fonts/fonts.conf**, **/etc/fonts/fonts.dtd** и **~/.fonts.conf**. Вот здесь бы и закончить обсуждение настройки *Xft* описанием этих файлов, сославшись на пресловутое **'man fonts-conf'**...

К сожалению, не получится. Хотя все современные рабочие среды по умолчанию используют *Xft*, параметры рендеринга они задают самостоятельно. При этом в «Центре управления KDE» или «Настройках шрифтов Gnome» прямых аналогий параметрам *fontconfig* может и не быть, хотя в целом соответствие найти не трудно. Нас будут интересовать следующие параметры:

ТАК ЧТО ТАМ ЗА TTF?

Фирме Apple Computer, Inc. принадлежит ряд патентов на технологию TrueType, которые, в частности, затрагивают реализацию интерпретатора, включаемого опцией **BYTECODE_INTERPRETER** в библиотеке TrueType. Поэтому использование библиотеки, включающей **BYTECODE_INTERPRETER**, будет законным лишь в том случае, если вы приобрели лицензию у Apple или находитесь вне зоны действия патентов США за номерами US5155805, US5159668, US5325479. Мы обратились к фирме Apple с вопросом, действуют ли указанные патенты на территории РФ и стран СНГ. К моменту подписания номера в печать ответа от Apple получено не было...

- **antialias** (сглаживание, отрисовка) – использовать или нет сглаживание глифов (элементов символов);
- **hinting** (уточнение, хинтинг) – использовать или нет уточнение;
- **hintstyle** (стиль уточнения): лёгкий, средний, полный;
- **autohint** – использовать или нет автохинтинг вместо обычного хинтинга. Можно бы и не упоминать: по умолчанию выключен во всех известных мне дистрибутивах;
- **rgba** – межточечное сглаживание (subpixel geometry): не использовать, **rgb**, **bgr**, **vrgb**, **vbgr**;

Возможности настройки *fontconfig* намного шире предлагаемых KDE/Gnome, поскольку позволяют задавать параметры для семейств шрифтов, отдельного шрифта, диапазона кеглей и даже определённого значения кегля. Типичный пример: отключение сглаживания для малых кеглей или, напротив, его включение для всех шрифтов семейства Vera. Вопрос только в том, захотите ли вы этим заниматься, тем более, если используете KDE, также позволяющий ограничивать сглаживание для диапазона кеглей. С другой стороны, к достоинствам Gnome следует безусловно отнести возможность «на ходу» менять dpi, моментально оценивая результат для конкретного шрифта и параметров сглаживания.

Насколько антиалиасинг улучшает вид текста, набранного сравнительно большим кеглем (текст страницы), и насколько, в то же время, он может быть неуместен (меню и панель навигации), демонстрирует очередной снимок экрана (рис. 4).

Из возможностей, предоставляемых исключительно *fontconfig*, стоит отметить задание состава семейства шрифтов. Обращаясь к услугам *Xft*, приложению, вообще говоря, совсем не обязательно указывать определённо существующий фонт. Можно требовать, скажем, sans-serif – и какой-то фонт из этого семейства *Xft* вам обязательно предоставит. А вот какой – определяется описанием этого семейства, выглядящим, например, так:

```
<alias>
<family>sans-serif</family>
<prefer>
<family>Tahoma</family>
<family>Verdana</family>
<family>Helvetica</family>
</prefer>
</alias>
```

Поскольку описание синтаксиса XML и тонкостей *fontconfig* выходят за рамки этой статьи, приходится адресовать интересующихся к уже упомянутому **'man fonts-conf'** и поиску в Сети по ключевым словам **fontconfig** или **fonts.conf**.

Шрифты и кегли

Чисто «техническая» сторона борьбы за лучший рендеринг после исправления libfreetype и выбора dpi заканчивается: выбор шрифтов, кеглей и параметров можно назвать сферой, где «на вкус и цвет товарищей нет». Поскольку «единого мнения» ожидать не приходится, хорошо бы договориться хотя бы о терминах.

Прежде всего – о кеглях. С учётом вышесказанного по поводу dpi очевидно, что размер на экране одной и той же "Tahoma 8" при dpi=75 и при dpi=120 будет различаться без малого в два раза, из чего следует, что нахваливая (или, напротив, уничтожающе критикуя) использование кегля определённого фонта, всегда нужно указывать dpi.

Во-вторых, восприятие одних и тех же шрифтов на экранах ЭЛТ- и ЖК-мониторов существенно отличается. Для последних нелишним будет оговорить, что монитор работает с рекомендованным разрешением (а оно у LCD только одно), а о работе ЭЛТ нельзя судить без указания частоты кадровой развёртки (refresh rate).

Механизмы рендеринга MS Windows и *X Window* не идентичны. С учётом варьирования dpi и зависимости от этого отображения различных фонтов, нет смысла гнаться за полной идентичностью рабочих столов. Некоторые TrueType-шрифты при особо малых кеглях могут вообще демонстрировать дефекты начертания, отсутствующие в MS Windows. Это, однако, исключение. В

большинстве случаев при «прочих равных условиях»: dpi, сглаживание (в MS Windows это называется “method to smooth edges of screen fonts”, выбор между “Standard” и “ClearType”, причём последний соответствует включённому сглаживанию), палитра, фонт, кегль), рабочие столы обеих графических сред весьма схожи.

При этом, однако, запуск различных программ под *X Window* опять-таки время от времени приносит болезненное разочарование. Причём, ещё до анализа отдельных приложений, несколько слов придётся сказать о графических инструментариях – наборах библиотек, средствами которых реализуется графический интерфейс как отдельных приложений, так и рабочих сред, вроде KDE или Gnome.

Gtk/Qt

В настоящее время большая часть графических приложений создаётся с использованием *Qt* (KDE, Opera) или Gtk (Gnome, *Gimp*, *Mozilla*). Они не единственные, но «заслуженный» *Tk* им не конкурент уже, а оригинальный *Fltk* – ещё. Поскольку использование *Qt* не обязательно предполагает KDE, а *Gtk* – Gnome, то естественно предположить, что должны существовать возможности настройки для приложений, которые используют названные инструменты вне KDE/Gnome. И такие средства есть: для Qt это **~/.qt/qtrc**, для Gtk – **~/.gtkrc-2.0**.

В qtrc вы можете обнаружить строку разрешения работы *Xft*, шрифт «по умолчанию», стиль и подробное описание палитр. Что касается **gtkrc-2.0**, то его может вообще не быть: в отличие от KDE, создающего qtrc при первом запуске, Gnome возможность существования **gtkrc-2.0** игнорирует. Создать такой файл бывает очень полезно – «не всё то Gnome, что на Gtk написано» .

Вообще, одновременное использование приложений, базирующихся на двух и более инструментариях, не прибавляет изящества рабочему столу: элементы одного назначения, но разного внешнего вида вряд ли кому-то нравятся. Не стремимся же мы к индивидуальному декорированию окон: win-менеджер может быть любой, но стиль отображения окон – всегда «один на всех». Реальность, однако, такова, что приверженцы KDE/Gnome испытывают друг к другу если не неприязнь, то жалость, и при этом существует ряд приложений, применяемых и теми, и другими. Прежде всего, это *OpenOffice.org*, *Gimp*, *Mozilla* «с потомками», *ROX Filer* и т.д.

Как может выглядеть приложение, использующее не тщательно подобранные вами в «Центре управления KDE» шрифты и кегли, а что-то совершенно чуждое? В большинстве случаев – плохо. Чтобы этого не было, укажите те же шрифт и кегль gtkrc-2.0, что используете в качестве «обычного» или «меню» . Например:

```
gtk-font-name = “Tahoma 8”
```

Элементы управления схожими не станут [для этого можно использовать специальные темы, например, *Gtk/Qt Engine* или QtCurve, – прим. ред.], но ROX с Firefox’ом в рамках KDE глаз станут резать гораздо меньше. Пользователи Gnome аналогичным образом могли бы поступить с *qtrc*, но Qt-приложений, незаменимых в среде Gnome, я практически не знаю. См. рис. 5.

Раз уж мы уделили столько внимания конфигурационным файлам, то укажем ещё два: **~/.gconf/desktop/gnome/font_rendering/%gconf.xml** и **~/.kde/share/config/kdeglobals**. Пути к ним однозначно указывают на то, что именно они конфигурируют, а в содержимом без труда угадываются «прямые потомки» параметров, обсуждавшихся выше.

Firefox, Thunderbird и все-все-все

К сожалению, и после настройки любимой среды и одновременно приложений, использующих «чужой» инструментарий, мы не застрахованы от появления на экране приложений с неудовлетворительно прорисованным текстом. Оставим в стороне графические пакеты: те, кто работает с графикой, сами должны знать, что такое dpi, и не только на экране... Не стоит обсуждать также фонты и кегль, которые приложение позволяет настроить: это, в сущности, ваше личное дело. Но что делать со слишком большими/маленькими или просто «корявыми» шрифтами в панелях, меню, подсказках?

В качестве примера предлагаю настроить *Firefox* (или любого другого «потомка» *Mozilla*). Среди настроек «Шрифты и цвета», кроме задания шрифтов и размеров, которые мы решили не обсуждать, имеется pop-up меню «Разрешение экрана». Как вы, вероятно, догадываетесь, никакое разрешение экрана *Firefox* не меняет, а меняет лишь многократно упоминавшийся dpi. Поскольку dpi, как мы выяснили, весьма «капризный» параметр, то если вы уж определили его для себя, то и *Firefox* заставьте придерживаться того же значения (выбор: «Системные настройки»).

Mozilla и все её потомки – [неполноценные] *Gtk*-приложения. Поэтому, если вы – приверженец KDE и ещё не создали **~/gtkrc-2.0**, то перед запуском *Firefox* самое время вспомнить об этом. Именно *gtk-font* по умолчанию использует *Firefox* в окне «Настройки». Если же один шрифт в панелях, меню и на кнопках вас не устраивает, то придётся прибегнуть к редактированию файла под названием **~/.mozilla/firefox/XXXX.default/chrome/userChrome.css**. Такого, правда, поначалу может и не быть, но заготовка для него под названием **userChrome-example.css** – имеется. Вот как задаётся шрифт большей части элементов интерфейса:

```
menubar, menubutton, menulist, menu,
menuitem {
font-family: “Tahoma” !important;
font-size: 8pt !important; }
```

Tahoma, в данном случае, указана не из-за каких-то её особых достоинств, а исключительно потому, что в соответствии с приведенными выше примерами именно её мы выбрали в качестве «первого» представителя семейства “serif”, шрифта приложений и шрифта по умолчанию для Gtk. Подобного единообразия рекомендую придерживаться и вам.

!important указывает на приоритет настройки относительно умолчаний. Более подробно о возможностях **userChrome.css** можно прочитать в **http://www.mozilla.org/unix/customizing.html**. Подобным образом userContent.css задаёт параметры вывода просматриваемых страниц. Однако, поскольку параметры вывода текста этих страниц можно менять и через «Настройки», мало кто этим пользуется.

И снова: не обольщайтесь. Авторы тем тоже знают, что такое каскадные таблицы стилей, и однажды поменяв тему, вы можете обнаружить, что шрифты опять «съехали». Обсуждение использования CSS-файлов в *Mozilla*, опять-таки, не тема настоящей статьи.

Итоги

Так могут ли шрифты под *X Window* выглядеть лучше, чем под XP? Наверное, иногда действительно могут, поскольку возможностей для «тонкой» настройки рендеринга *X Window* предоставляет явно больше. Разнообразие возможностей, однако, хотя и является источником прогресса, всегда способствует поддержанию некоторого уровня беспорядка (и это ещё мягко сказано). Приятное же глазу единообразие создать можете либо вы сами, либо автор вашего дистрибутива. Причём во втором случае, «дистростроитель» должен обеспечить унификацию настроек предлагаемых им приложений, а вам рекомендуется использовать пакеты исключительно своего дистрибутива.

Настойчивый и компетентный пользователь всегда может добиться желаемого результата. С другой стороны, Linux-хиты последнего времени, такие, как Ubuntu/Kubuntu, выглядят в сравнении с XP вполне достойно. Будем надеяться, что и впредь программисты *X.Org* будут развивать систему, используемую в качестве основы для различных моделей рабочих столов. Разработчики графических сред смогут предложить исчерпывающе полные реализации «пользовательских окружений». А составители дистрибутивов постараются избавить «простых» пользователей от многочисленных настроек – с одной стороны, и разочарований при сравнении с Windows XP – с другой. **LXF**

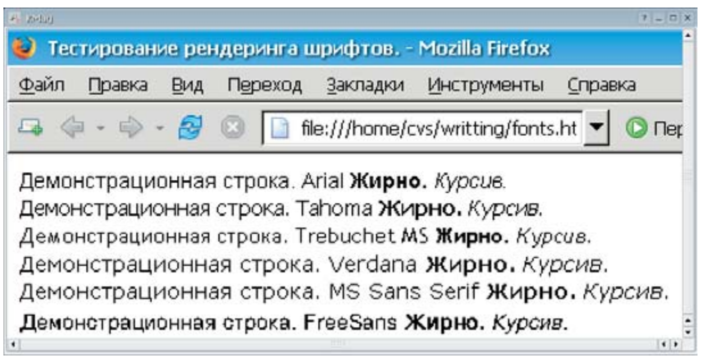
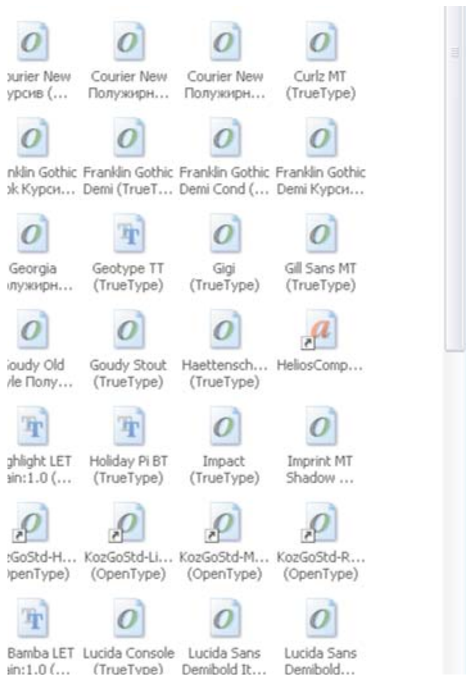


Рис. 5. Шрифты «как в Windows» – финальная версия.





ДРАЙВЕР СЕТЕВОГО УСТРОЙСТВА – своими руками

ЧАСТЬ 2 Зарегистрировать пакет данных – это даже не половина, а четверть дела. **Игорь Тимошенко** продолжает мастер-класс по написанию сетевых драйверов Linux

МЕСЯЦ НАЗАД

Мы посмотрели на ядро Linux изнутри и написали скелет сетевого драйвера, сохранявшего пакеты в файле журнала.

НА ДИСКЕ

• Полные исходные тексты обсуждаемого в статье драйвера

В прошлый раз мы написали простейший драйвер, который определялся в системе как сетевой, инициализировался как интерфейс и, пользуясь доверием ОС, принимал направленные на него пакеты. На этом его лояльность сетевой подсистеме Linux заканчивалась, и он не отдавал их обратно в сеть, а сохранял в файле **/var/log/messages**, где мы и изучали их, используя текстовый редактор и утилиту *tail*. Такое поведение не даёт нам оснований считать программу законченной, и сейчас мы продолжим работу над ней, чтобы довести её до уровня «взрослых» сетевых драйверов.

Для выполнения поставленной задачи мы должны, прежде всего, дать нашему драйверу возможность переправлять принятые пакеты в буфер сетевой подсистемы. Рассмотрим, как это обычно происходит в Linux.

Все передаваемые и принимаемые пакеты в сетевой подсистеме Linux хранятся и обрабатываются в специальных типовых буферах, называемых структурами буфера сокетов (**struct sk_buff**). Имеющиеся в памяти буферы организованы в очередь, которая управляется ОС. В случае, если нужно разместить поступивший в сетевую подсистему пакет, а существующие буфера заполнены данными, система выделяет в памяти место под новый буфер. Когда пакет обработан и надобности в нём больше нет, буфер не уничтожается, а помечается как свободный и может быть использован для размещения вновь поступившего пакета. Таким образом, обычно в системе существует некоторое количество свободных буферов, которые могут быть использованы для обмена пакетами с сетевыми интерфейсами.

После получения пакета драйвер должен запросить у ОС указатель на свободный буфер для его размещения. Делается это с помощью функции **dev_alloc_skb()**, которой в качестве параметра передаётся длина полученного пакета. Функция возвращает указатель на свободный буфер, драйвер должен заполнить его данными. Для копирования данных в буфер удобно использовать функцию **memcpy()** совместно с **skb_put()**, которая, при необходимости, расширяет буфер до нужной длины и согласует это с системой. Для того, чтобы ОС могла правильно интерпретировать полученные данные в структуре буфера, нужно определить поле **unsigned short protocol**, содержащее код MAC-уровня, указывающий на тип инкапсулированного в пакет сетевого протокола. Обычно в сетях Интернет используется IPv4, а соответствующий ему код содержится в макроопределении ETH_P_IP в заголовочном файле **linux/if_ether.h**, на который ссылается **linux/netdevice.h**. После того, как структура заполнена, об этом нужно сообщить системе при помощи функции **netif_rx()**, в качестве аргумента которой передаётся указатель на заполненный буфер. Изложенную последовательность действий удобно объединить в отдельную функцию (мы назовем ее **load_pack()**), которой передается указатель на буфер с принятым пакетом:

```
/* Структура буфера данных */
struct my_buf {
int tlen;
unsigned char tail[2];
        union tbuffer{
                unsigned char tbuff[1024];
                struct iphdr thdr;
        } tb;
};
static void load_pack(struct my_buf *xbf) {
        struct net_device_stats *stats = ssl_dev.priv;
        struct sk_buff *r_skb;

        r_skb = dev_alloc_skb(xbf->tlen);
        if (r_skb == NULL) {
                printk(KERN_WARNING «ssl: memory squeeze,
dropping packet.\n»);
                stats->rx_dropped++;
                return;
        }
        r_skb->dev = &ssl_dev;
        memcpy(skb_put(r_skb, xbf->tlen), xbf->tb.tbuff, xbf->tlen);
        r_skb->protocol=htons(ETH_P_IP);
                netif_rx(r_skb);
        stats->rx_bytes += xbf->tlen;
        stats->rx_packets++;
}
```

Следующий шаг – привязка драйвера к аппаратному устройству, управляющему передачей данных на физическом уровне – по проводам в виде электрических сигналов. Обычно этим занимаются телефонные модемы или сетевые карты. В современных компьютерных сетях данные могут передаваться также и по оптическим кабелям, и даже вообще без кабелей – по радиоканалу, но это – темы для отдельного разговора. Мы не будем сейчас разбираться в существующих стандартах передачи данных, а выберем в качестве аппаратного устройства асинхронный адаптер последовательного порта (см. врезку), который представлен в системе файлами устройств **/dev/ttyS(0….n)** и попытаемся связать два компьютера в сеть по кабелю.

Для дальнейшей работы над драйвером нам потребуется нуль-модемный кабель, который можно недорого приобрести (как ни странно) почти в каждом компьютерном магазине. При помощи кабеля нужно соединить порты COM1 и COM2 вашего компьютера (или двух компьютеров). Все действия для разных портов нужно проводить в разных текстовых консолях. Если вы проводите наши эксперименты в графической среде, то сможете разместить консоли на одном экране.

Программирование режима работы адаптера удобнее производить в несколько этапов. При загрузке системы адаптеры COM-портов обычно инициализируются для работы в подходящем режиме, и для работы нужно только установить желаемую скорость обмена. Для порта COM1 это можно сделать командой:

```
stty -F /dev/ttyS0 115200
```

Порту COM2 будет соответствовать файл устройства **/dev/ttyS1**.

При загрузке адаптеры программируются для работы в режиме прерываний. Поскольку мы не намерены использовать этот режим (равно как и буферизацию), его нужно отключить. Для этого при инициализации интерфейса, в функцию **ssl_open()**, нужно вставить следующие строки:

```
outb_p(0, BASE + ICR_N);        /* запрет прерываний */
outb_p(0, BASE + IIDR_N);       /* запрет буферизации */
reg_mcr.byte = 0;               /* Сброс дополнительных
сигнальных линий */
```

где BASE, ICR_N, IIDR_N – определенные нами в исходном коде драйвера директивы препроцессора, упрощающие работу с последовательным портом.

Теперь нам остаётся модифицировать функцию передачи **ssl_xmit()** таким образом, чтобы передаваемый пакет сразу направлялся в адаптер. Изменённая функция будет выглядеть следующим образом:

```
static int ssl_xmit (struct sk_buff *skb, struct net_device *dev) {
        if (skb->len > 1024 – 3) {
                printk(KERN_WARNING «ssl: T_buffer is small,
dropping packet.\n»);
                stats->tx_dropped++;
        } else {
                /* Передача пакета в COM-порт */
                shift = 0;
                while (shift + 1 <= skb->len) {
                        while ((inb_p(BASE + LSR_N) &
0x20)==0);
                                outb_p(skb-
>data[shift], BASE);
                                shift++;
                        }
                dev_kfree_skb(skb);
                stats->tx_bytes += skb->len;
                stats->tx_packets++;
        }
return 0;
}
```

Чтобы проверить работоспособность получившегося драйвера, нужно соединить кабелем передающий COM-порт вашего компьютера с

>>

АСИНХРОННЫЙ АДАПТЕР

Асинхронный адаптер в компьютере существует в виде микросхемы (UART 16550A), управляющей передачей данных по кабелю, обычно – в стандарте RS-232. Кабель подключается к разъёмам портов COM1 и COM2, которые находятся на задней стороне системного блока. В каждом из направлений, данные передаются по сигнальному проводу (TX->RX) за счёт изменения электрического напряжения на выходе передатчика относительно уровня «земли». Для управления передачей в кабеле имеется несколько дополнительных линий, из которых для нас интересны RTS->CTS (используются для аппаратного управления потоком данных) и DTR->DSR (сигнал подтверждения связи адаптера с каким либо внешним устройством, например – с модемом). Поскольку передача данных обычно бывает двухсторонней, в кабеле предусмотрен полный набор сигнальных линий для каждого из направлений.

Адаптер управляется программно. Для этого у него имеются регистры, доступные для чтения-записи из адресного пространства ввода-вывода компьютера. Перед началом работы адаптер необходимо запрограммировать в нужный режим работы путём занесения соответствующих данных в его управляющие регистры. Данные передаются словами, обычно – байтами. Для передачи байта нужно записать его в регистр данных, после чего адаптер начнёт преобразовывать его в электрические сигналы. Об окончании передачи можно узнать, прочитав содержимое регистра состояния линии. По окончании передачи в регистр данных можно записывать следующий байт. Из содержимого регистра состояния линии можно так же узнать о наличии принятого байта, который можно получить, прочитав регистр данных, после чего адаптер будет готов принять следующий байт. Состоянием дополнительных сигнальных линий можно управлять путём записи данных в регистр управления модемом, а анализировать их – читая содержимое регистра состояния модема.

В адаптере предусмотрена возможность генерации сигналов прерывания по окончанию передачи и по приёму очередного байта, а также передача сразу нескольких байт через встроенный буфер, но мы не будем рассматривать эти режимы.

Процедура программирования асинхронного адаптера подробно описана в статье, которую можно найти по адресу: **http://control-m.maket.net/rs-232/ch2.html**.

СИСТЕМНЫЙ ТАЙМЕР

Системный таймер – это устройство, генерирующее аппаратное прерывание IRQ 0 с заданной частотой – HZ раз в секунду. Значение HZ зависит от назначения системы (рабочая станция, сервер...), архитектуры (i386, PPC, ...), указывается при конфигурации ядра и затем сохраняется в **/usr/include/asm/param.h**. Обычно оно равно 1000. Каждый раз, когда ядро вызывает обработчик прерывания IRQ 0, он увеличивает значение глобальной переменной **jiffies**, которая, таким образом, является точкой отсчета при указании временных интервалов в ядре. Помимо этого, прерывание системного таймера используется Linux для реализации механизма таймеров ядра (kernel timers). Таймеры ядра обеспечивают выполнение некоторых функций (называемых обработчиками таймера – **timer handler**) по истечении определенного промежутка времени и описываются структурой **timer_list**. Для управления таймерами ядра служат специальные системные функции. Не путайте функцию-обработчик тайме2ра ядра с функцией-обработчиком прерывания системного таймера – это совершенно разные вещи!



<< принимающим COM-портом второго компьютера. Для программирования порта на принимающей стороне можно использовать следующий сценарий:

```
#!/bin/sh
#Подставить нужный номер
PORT=/dev/ttyS1
stty -F $PORT cs8 -isig -icanon -iexten -opost -ixon -icrnl -hupcl -echo crtscts -parenb 115200
stty -a < $PORT
```

После чего можно отображать на консоль принимаемые данные с помощью команды

```
cat /dev/ttyS1
```

Затем, если вы скомпилируете и запустите наш драйвер обычным образом, то сможете наблюдать на консоли прослушивания появление примерно таких картинок:

```
!»#$%&'()*+,-./01234567ET
@ЫКю юЦхs0
C /D=
!»#$%&'()*+,-./01234567ET
@ЫЈю ю! s0
D /DЧ%
```

Это, конечно, не так информативно и красиво, как было раньше, но вполне подтверждает, что наш драйвер уже научился отправлять данные на физический уровень.

Следующий шаг в развитии драйвера – прием данных из кабеля через адаптер последовательного порта и передача их в сетевую подсистему Linux. Для того, чтобы принять байт данных из адаптера, нужно просматривать содержимое регистра состояния линии, а именно – состояние его младшего бита. Если он установлен в **1**, это означает, что очередной байт принят из линии и находится в буферном регистре приёмника. Буферный регистр доступен через регистр данных адаптера. Чтение принятого байта освобождает буфер для приёма следующего. Следовательно, принимающая функция драйвера должна проверять младший бит регистра состояния и всякий раз, когда он установится в **1**, считывать принятый байт. Для решения этой задачи нам следовало бы воспользоваться механизмом аппаратных прерываний. Но, поскольку мы сразу договорились, что для упрощения программы не будем использовать прерывания асинхронного адаптера, то воспользуемся (весьма опосредованно) прерыванием системного таймера, а точнее – механизмом таймеров ядра (см. врезку «Системный таймер»).

Приступим к постепенному воплощению этой идеи. Для начала, подключим к программе все необходимые заголовочные файлы напишем «скелет» обработчика таймера, а затем – реализуем в нем побайтовый приём данных из адаптера и синхронизацию приёма-передачи пакетов для взаимодействующих адаптеров.

```
#include <linux/timer.h>
#define T_TICK HZ/100
struct timer_list timer;
spinlock_t timer_lock = SPIN_LOCK_UNLOCKED;
/* ОБРАБОТЧИК ТАЙМЕРА */
void ssl_tick (unsigned long data) {
        spin_lock(&timer_lock);
        ...
        mod_timer(&timer, jiffies + T_TICK);
        spin_unlock(&timer_lock);
}
```

Переменная **timer** и есть наш таймер ядра, а вот **timer_lock**, наверное, требует некоторого пояснения. Мы собираемся организовать прием данных, передаваемых асинхронно, и поэтому нам важно исключить одновременный (конкурентный) доступ к буферу **my_buf** и другим внутренним переменным. Эту синхронизацию и обеспечивает **timer_lock**. Мы захватываем блокировку перед входом в функцию-обработчик (вызов **spin_lock**) и освобождаем при выходе из нее (вызов **spin_unlock**). Если код, выполняющийся на другом процессоре в SMP-системе, попробует захватить уже взятую нами блокировку, он не сможет двинуться дальше до тех пор, пока мы не освободим ее, вызвав **spin_unlock**. В конце обработчика вызывается функция **mod_timer()**, которая обеспечивает повторное выполнение данного кода через T_TICK прерываний системного таймера. Это необходимое действие – таймеры ядра обеспечивают лишь однократное срабатывание.

Для начальной инициализации работы таймера нужно также модифицировать функцию **ssl_open()**:

```
int ssl_open (struct net_device *dev) {
        printk(KERN_WARNING «ssl: ssl_open called.\n»);
        netif_start_queue (dev);
        /* инициализация COM1 */
        outb_p(0, BASE + IIDR_N);            /* запрет
буферизации */
        outb_p(0, BASE + ICR_N);             /* запрет
прерываний */
        reg_mcr.byte = 0;                    /* Сброс
дополнительных сигнальных линий */
        /* инициализация таймера */
        init_timer(&timer);
        timer.expires = jiffies + HZ;
        timer.function = ssl_tick;
        add_timer(&timer);
        return 0;
}
```

Для корректного завершения работы таймера после остановки интерфейса нужно также немного модифицировать функцию **int ssl_stop ()**:

```
int ssl_stop (struct net_device *dev) {
        del_timer_sync(&timer);  /* остановка таймера */
        printk (KERN_WARNING «ssl: ssl_stop called.\n»);
        netif_stop_queue(dev);
        return 0;
}
```

Теперь можно приступать к следующему этапу – приему данных. Для этого нам нужно организовать в обработчике прерывания побайтный приём данных из адаптера и решить проблему синхронизации пакетов данных при передаче. Проблема заключается в том, что в процессе побайтной передачи нужно как-то отмечать, где заканчивается один пакет и начинается другой. Это можно делать двумя способами: программно и аппаратно.

НЕ ДЕЛАЙТЕ ЭТОГО ДОМА, ДЕТИ!

Приведенный в статье код драйвера является полностью рабочим. Однако необходимо учесть, что он создавался с учебными целями, а потому умышленно прост и демонстрирует очень фамильярное отношение к ресурсам компьютера и системы. Его нельзя рассматривать как образец правильного дизайна драйвера для Linux. Если вы всерьез интересуетесь разработкой собственных драйверов, рекомендуем вам обратиться к специальной литературе, например, «Linux Device Drivers, 3rd Edition» (издательство O'Reilly) и «Linux: сетевая архитектура. Структура и реализация сетевых протоколов в ядре». (издательство «Кудиц-Образ»).

В случае программной синхронизации в конце каждого пакета размещается некая сигнальная последовательность символов. Это подразумевает кое-какие неудобства – нужно гарантировать, что сигнальная последовательность не встретится в теле пакета или мириться с некоторым количеством ошибок, поэтому в нашем драйвере мы реализуем аппаратную синхронизацию. Для этого мы задействуем дополнительные сигнальные линии, изменение логического состояния в которых обозначает окончание передачи пакета. Перед началом передачи передающее устройство устанавливает в синхронизирующей линии логическое состояние, обозначающее начало пакета, затем передаёт сам пакет, после чего сбрасывает сигнал. Принимающая сторона следит за этим сигналом, принимает данные, пока он установлен и прекращает приём при его сбросе, после чего начинает обрабатывать принятый пакет. Если передатчик в это время начнёт передачу следующего пакета, приёмник не сможет его принять (буфер ещё занят предыдущим пакетом) и, таким образом, пакет будет потерян. Чтобы избежать этого, нужно использовать встречный сигнал синхронизации, который должен устанавливаться приёмником, когда он готов к приёму, и отслеживаться передатчиком.

Мы задействуем для этих целей две пары линий: DTR->DSR и RTS->CTS. Первая пара отводится передатчику, вторая – приемнику. Теперь мы готовы к тому, чтобы привести полный текст обработчика таймера.

```
/* ОБРАБОТЧИК ТАЙМЕРА */
void ssl_tick (unsigned long data) {
        struct net_device_stats *stats = ssl_dev.priv;
        struct my_buf *qbf = &xbf;
        spin_lock(&timer_lock);
        /*ПРЯМОЕ ЧТЕНИЕ ИЗ ПОРТА ttyS0 УПРАВЛЕНИЕ RTS->
DSR<-*/
        reg_msr.byte = inb_p(BASE + MSR_N);
        if (reg_msr.bit_reg.dsr != 0) {
/*проверяем запрос от передатчика */
                reg_mcr.bit_reg.rts = 1;
                outb_p(reg_mcr.byte, BASE + MCR_N);
/*подтверждаем готовность приёма*/
                shift = 0;
                while (reg_msr.bit_reg.dsr == 1) { /*принимаем
пока передают*/
                        /*ждём установки бита
готовности приёмника*/
                        while ((inb_p(BASE + LSR_N) &
1)==0);
                        /*читаем очередной байт*/
                        qbf->tb.tbuff[++shift-2] = inb_
p(BASE);

                        if (shift > 1024) {
                                printk(KERN_
WARNING «ssl: R_buffer is small and overfull,
                        dropping packet.\n»);
                                stats->rx_dropped++;
                                goto r_out;
                        }
                        reg_msr.byte = inb_p(BASE +
MSR_N);
                } /* Пакет принят */

                if (shift == 0) {
                        printk(KERN_WARNING «ssl:
Packet is 0 byte, dropping packet.\n»);
                        stats->rx_dropped++;
                        goto r_out;
                }
                qbf->tlen = shift-2;
                //tpdumpk(&xbf);        /*Дамп принятого в
буфер пакета в логе*/
                load_pack(&xbf);        /*Отправка
принятого пакета в SKB*/
        } //else printk(KERN_WARNING «ssl: Line DSR is free.\n»);
r_out:
        reg_mcr.bit_reg.rts = 0;
        outb_p(reg_mcr.byte, BASE + MCR_N); /*снимаем
подтверждение приёма*/

        mod_timer(&timer, jiffies + T_TICK);
        spin_unlock(&timer_lock);
}
```

Отмеченные комментариями вызовы функций удобно использовать при изучении работы драйвера и при отладке.

Передача пакета происходит по такому же принципу. Перед началом передачи выставляется сигнал DTR, и передатчик ожидает готовности приёмника в течении времени T_TICK. Если сигнал готовности получен, происходит побайтовая передача данных. Каждый следующий байт передаётся только после установки бита готовности передатчика (**out_ready**) в регистре состояния линии (LSR). По окончании передачи пакета, передатчик снимает сигнал DTR и приёмник заканчивает цикл приёма пакета. Полный текст программы можно найти на прилагаемом диске.

Следует отметить особенность работы адаптера UART 16550A. При последовательной передаче, два байта «зависают» в его буфере, и их необходимо «проталкивать», записывая «пустые» символы в конец пакета. Эта особенность учтена в приведённой программе. «Пустые» символы помещаются в раздел **unsigned char tail[2]** буфера приёмника (**my_buf**).

Мы получили «настоящий» драйвер для передачи IP-пакетов через COM-порт. Для его тестирования потребуется второй компьютер. При использовании драйвера помните, что он расчитан на работу с портом COM1 – для использования COM2 и т.п. Нужно изменить значение BASE и пересобрать модуль. Для тестирования драйвера можно использовать сценарий настройки, приведенный в первой части статьи. Выполните его на обоих компьютерах, предварительно изменив на втором узле IP-адрес на 192.168.0.2. После этого можно проверить связь командой ping.

Получившийся драйвер, как и любую сделанную вещь, можно улучшить. Прежде всего, можно отказаться от использования дополнительного буфера при приёме пакетов. Это немного упростит программу и уменьшит время обработки. Кроме того, можно принимать данные по собственному прерыванию адаптера. В этом случае можно также воспользоваться режимом буферизации, который позволяет принимать по несколько байт за одно прерывание. Все эти меры могут существенно увеличить быстродействие программы и уменьшить нагрузку на процессор, но это уже тема для отдельного разговора. LXF



Учебники >>

Наши эксперты помогут вам с любым приложением Linux

ЧЕМ БЫТЬ?



Евгений Балдин подумывает, не запустить ли серию статей про TeX!

Жизнь такая, какая она есть. Но так уж устроен человек, что ему хочется понять «почему»? Например, «Почему Linux?» И вообще, «Какой он породы?» «Разведка доносит», что он сильно похож на Unix, но на снимках экрана вылитый «Уиндовз». Тем более, что сравнивать приходится с известным.

«ИЗ FIDO7.RU.OS.CMP:
Q8: А ГДЕ ЭТО ЕСТЬ И КРУЧЕ?
A8: В VMS!»

Так как рабочий стол большинство представляет одним способом, то получаем именно то, о чём вы сейчас подумали. А на сколько это необходимо? Надо ли так старательно копировать внешние признаки? Может быть озадачиться техническим превосходством?

У той же VAX/VMS от DEC было масса хороших идей. Эта зрелая система создавались позже Unix, с оглядкой на него. Там было решено много проблем до сих пор присутствующих в Linux. Недавно в новостной группе comp.text.tex был вопрос от человека, который использует эту систему 18 (!!!) лет и не собирается с неё уходить. Всё его там устраивает, только LaTeX посвежее нужен. Может быть лучше прикинуться именно этой системой?

Linux это не копия — это «котёл для идей». Он всё переварит, но лучше чтобы ингредиенты были съедобные.

КОД В ЭТОМ РАЗДЕЛЕ

Печатать в журнале код не так легко, но мы надеемся, что некоторые простые правила позволят избежать недоразумений. Если строка кода оказывается слишком длинной, она переносится на следующие, причем все они обрамляются синим прямоугольником:

```
procedure
TfrmTextEditor.mniWordWrapClick
(Sender: TObject);
```

В противном случае, между строками будет небольшой промежуток:

```
 mniWordWrap.Checked := false
end;
```

Как правило, весь код можно найти на прилагаемом к журналу диске.

В ЭТОМ МЕСЯЦЕ ИЗУЧАЕМ...

СОВЕТ МЕСЯЦА CONVERT И MOGRIFY

ImageMagick неисчерпаем, как и атом. Распространенный повсеместно – от KDE-приложений до PHP-сценариев, этот многофункциональный инструмент для обработки изображений содержит батальон мелких утилит. Их насчитывается более сотни, но две самых полезных – *convert* и *mogrify* – могут потягаться даже с *Gimp*. С их помощью можно изменить размеры, размыть, кадрировать, повысить четкость и преобразовать изображение, которого вы и в глаза не видели. Отличие между convert и mogrify состоит в том, что последний позволяет перезаписывать оригинальные файлы вместо того, чтобы создавать новые.

Несмотря на всю функциональность, естественная область применения *convert* – изменение формата изображения. Вам потребуется указать входной и выходной файлы – только и всего. Convert позаботится о деталях. Следующий пример превращает JPG в PNG:

```
convert pic.jpg pic.png
```

ImageMagick идеален для пакетной обработки изображений, но использование шаблонов может дать неожиданный результат. Например, **convert *.jpg *.png** просто не заработает. Решение состоит в использовании несколько другого синтаксиса и замене *convert* на *mogrify*.

Вместо шаблона нужно просто указать формат выходного изображения. Чтобы превратить все JPEG в PNG, наберите:

```
mogrify -format png *.jpg
```

Такой подход можно использовать с любыми параметрами *convert* и *mogrify*. Подставьте вместо **format** опции **crop** или **scale**, если вам нужно откадрировать или отмасштабировать изображение и так далее.



LINUX ДЛЯ НАЧИНАЮЩИХ

Audacity Запись и редактирование

Слушайте повелителя подкастов **Энди Ченнела (Andy Channelle)** и учитесь записывать, править и микшировать звук.



МЕСЯЦ НАЗАД

Мы познакомились с электронной почтой, календарем и прочими полезными вещами в Evolution – лучшем менеджере личной информации от Novell.

Шляпы долой перед Рики Жерве (Ricky Gervais). Простой человек из Ридинга создал отличное ТВ-шоу The Office, сыграл злодея в Alias и подарил нам самый популярный подкаст в (совсем короткой) истории этого сервиса (вы можете найти его на **www.guardian.co.uk/rickeygervais**). Если вы хотите повторить успех этого человека, вам понадобится несколько вещей: артистические способности, хороший сценарий и… инструменты и технические навыки для записи звука и его вещания в Интернете.

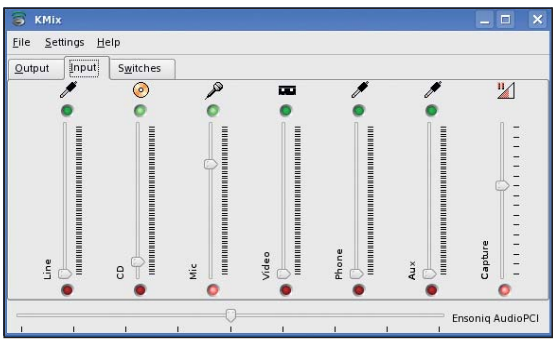
Да, в этом месяце мы поговорим о записи и редактировании звука, который вы можете использовать для подкастинга или вставить в свою видео-презентацию. Мы рассмотрим *Audacity* – свободное приложение, которое не уступает своим платным конкурентам. Изначально *Audacity* предназначался для музыкантов, но в последнее время его все чаще применяют как лучшее приложение для подкастинга на всех основных платформах.

Для начала вам потребуется раздобыть и установить эту программу. Большинство пользователей справятся с этой задачей в два счета, используя свои менеджеры пакетов (*Yast*, *Yum*, *Apt* и другие), но если таким путем установить *Audacity* не получается, вам поможет сайт проекта (**www.audacity.sourceforge.net**), где можно найти бинарные сборки для большинства дистрибутивов – а для искателей приключений и исходный код! Если почему-то все перечисленные способы не помогли, то вам может пригодиться сайт *Rpmfind* (**www.rpmfind.net**); по запросу *Audacity* вы получите список пакетов для различных дистрибутивов. Также вам понадобится *Lame* для кодирования звука в формат MP3, который сможет слушать практически любой пользователь. Некоторые компании включают пакет с *Lame* в состав своих дистрибутивов, но из-за возможности судебных разбирательств большинство поставщиков так не делают, поэтому пользователям приходится искать и устанавливать *Lame* самостоятельно. Поиск по ключевым словам 'Lame RPM' плюс название дистрибутива или 'Lame DEB' поможет вам найти нужный файл.

Любой аудио-проект можно разделить на 4 составные части: захват, редактирование, сведение и вывод. Обычно второй и третий пункты на практике сливаются в один, но при работе с аудиовещанием или звуковым сопровождением видео это разные понятия. Начнем по порядку – займемся записью звука на компьютер.

Подготавливаем звуковую систему

Я буду создавать собственный подкаст, на который меня вдохновила недавняя ностальгическая телепередача про *Jackanory*. В подкасте ожидается чтение стихов Роберта Браунинга (Robert Browning) «Дудочник

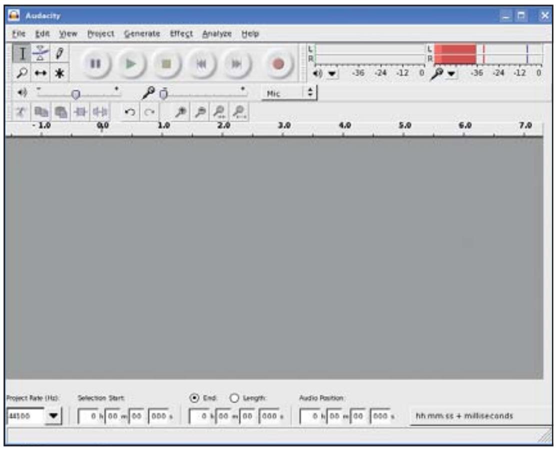


Kmix **– это фирменный регулятор громкости в составе KDE.**

в пёстром костюме из Гамельна» – «Pied Piper of Hamelin» (авторские права не нарушаются) под соответствующий музыкальный аккомпанемент, который будет «оживлен» несколькими звуковыми эффектами. Вы можете последовать моему примеру. Подразумевается, что у вас есть необходимый минимум: настроенная звуковая карта, колонки и микрофон. Вы будете осуществлять запись через другое устройство (музыкальный центр), в этом случае сигнал подается на ПК через разъем 'line in'. Если вы будете пользоваться именно этим способом, в дальнейшем замените все упоминания 'mic' на 'line in'.

Сперва проверим звуковую систему, открыв *Kmix* или *Регулятор громкости* на панели задач. Ссылки на эти микшеры обычно находятся в главном меню в разделе Мультимедиа. Запущенный микшер отображается в виде небольшого значка с рупором в системном лотке (небольшой области на панели задач, где размещаются запущенные апплеты). Итак, откройте микшер и, перейдя на вкладку Input, включите канал Mic (или Line in). Убедитесь, что оба индикатора – зеленый и красный – включены, и что бегунок находится ближе к верху. Если все включено, то при щелчке по микрофону вы услышите звук из колонок.

Есть и другой способ проверить сигнал: запустите *Audacity*, убедитесь, что на панели Recording (запись) выбран микрофон, и выберите Start Monitoring из выпадающего меню рядом с иконкой микрофона в панели Monitoring. Теперь аудиометр в *Audacity* будет реагировать на стук по микрофону или речь, визуально отображая сигнал.



Протестировав звуковую систему, переходим к записи. Audacity готова взяться за дело…

Тише, идет запись!

До того, как мы приступим к записи, имеет смысл зайти в настройки программы (Правка > Настройки – Edit > Preferences) и задать количество используемых каналов записи равным двум – так мы получим стереозвук. Также здесь нужно зайти на вкладку форматов файлов (File Formats) и указать программе на расположение библиотеки *Lame* (моя подсказка: это то место, где вы ее сохранили). Теперь можно начинать.

Пара тестовых записей позволит нам оптимально расположить микрофон, подстроить уровень сигнала и избавиться от шума на заднем плане, поэтому нажимайте кнопку Запись (Record) на панели управления звуком и говорите, пойте и делайте все то, что вы хотите записать. Если все идет правильно, то линия записи начнет двигаться слева направо, «рисуя» звуковой спектр в реальном времени. Если уровень записи слишком низок, то рисунок может быть близок к центру полосы и наоборот: на большой громкости линии будут выходить за пределы полосы. В идеале, нужно добиться промежуточного результата, когда самые громкие участки по высоте приближаются к верхней и нижней границам полосы.

Вы могли уже заметить, что при каждом нажатии на кнопку Запись создается новая дорожка, которая исполняется параллельно с уже записанными. Это полезно при работе с несколькими взаимодействующими дублями, но только в этом случае вам нужно снова заглянуть в настройки и удостовериться что пункт «Проигрывать другие дорожки при записи новой» (Play Other Tracks While Recording The New One) в разделе «Ввод/вывод звука» (Audio I/O > Playthrough) включен. Между прочим, таким же способом можно использовать *Audacity* в качестве мультитрекового инструмента записи звука. Если вам не нужна эта функция, просто закройте предыдущую дорожку, щелкнув по маленькому крестику в ее верхнем левом углу.

Теперь пора за дело. Не стесняйтесь делать несколько попыток – это ведь репетиция и возможность не только доработать сценарий, но и опробовать различные варианты подачи текста. Сохраняя каждый дубль, даже тот, который поначалу показался вам неудачным, вы можете разнообразить ваш проект при окончательном сведении, составив его из разных дублей. Кстати, если вы записываете несколько дублей, позаботьтесь о том, чтобы в каждом случае вы были на одинаковом расстоянии от микрофона. Перекосы громкости можно исправить, но гораздо лучше иметь приличный материал изначально.

Audacity хранит свои временные файлы в виде несжатых WAV-файлов, что нельзя назвать рациональным расходованием дискового пространства – каждая минута записанного звука отнимает 10 Мб. Тем не менее, если ваш диск не заполнен под завязку, я рекомендую сохранять все дубли (или хотя бы их полезные фрагменты) и хранить их до того, как вы полностью сведете весь звук.

Если для на вашем жестком диске мало места, есть смысл сохранять дубли в формате MP3 или OGG. В обоих случаях 1 минута звука будет соответствовать приблизительно 1 Мб, однако, при равном размере OGG дает лучшее качество, чем MP3. Для того, чтобы экспортировать дорожку или ее фрагмент, выберите Файл > Экспортировать в MP3 (или в Ogg Vorbis) и сохраните файл в нужном месте. Когда вам понадобится эта запись в следующий раз, выберите пункт Проект > Импортировать звуковой файл. Должен вас предупредить: при многократном экспорте/импорте с использованием сжатия с потерями, качество звука может заметно ухудшиться, так как в процессе сжатия из звукового спектра удаляется часть информации. Из-за этого я рекомендую по возможности использовать для хранения несжатые форматы.

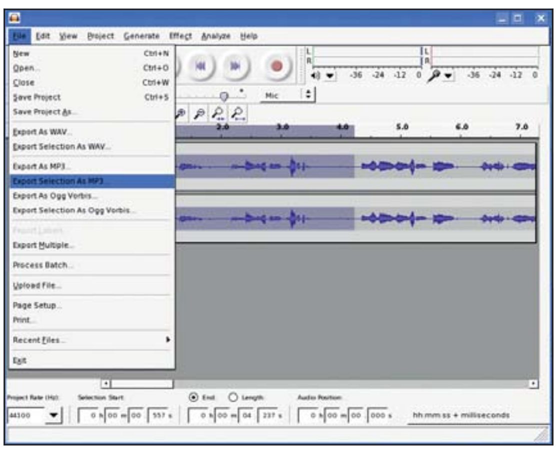
Редактируем записанное

Под термином «редактирование» я понимаю в данном случае чистку, применение спецэффектов и, в целом, подготовку файлов к сведению. Только теперь мы наконец познакомимся с интерфейсом *Audacity*.

При работе со звуком на первом месте стоит чистка шумов. Это значит, что мы должны выделить все паузы и вставить на их место участки тишины. Чтобы выделить звуковой фрагмент, щелкнете левой кнопкой мыши по началу тихого участка и перетащите указатель до его конца.

Теперь выберите Создание > Тишина (Generate > Silence...) и нажмите кнопку Создать. Для достижения профессионального качества пауз делайте длину участка с тишиной несколько меньше необходимого. Таким образом, у вас получатся небольшие зазоры до и после паузы, и вы сможете сделать плавные переходы от звука к тишине и наоборот. Выделив такой зазор, используйте опцию Эффекты > Плавное нарастание или Эффекты > Плавное затухание (Effects > Fade In/Out), после чего запись будет звучать лучше, без резких обрывов.

>>



Выделенный фрагмент можно экспортировать в различные форматы. WAV ориентирован на профессионалов, MP3 является наиболее распространенным, а OGG предоставляет оптимальный баланс между качеством и размером файла.

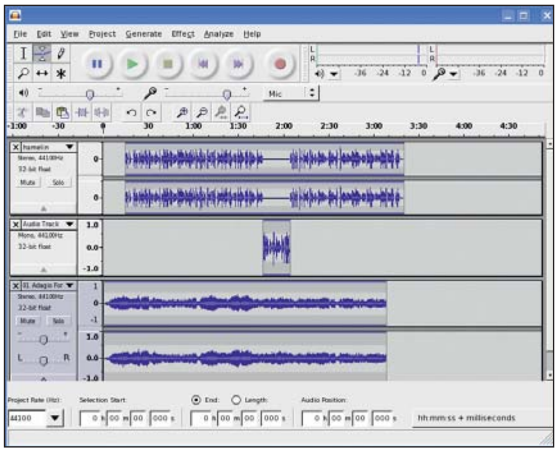
ПОДСКАЗКА

• В левой части каждой дорожки имеется полезная информация, включающая формат записи (32 бита с плавающей точкой – стандарт для стерео-дорожки) и частоту дискретизации (по умолчанию это 44,1 Кгц). Эта информация всегда на виду, но она, как правило, редко кому нужна. Если вы пользуетесь самой новой версией *Audacity* – 1.3, то под этими данными вы найдете небольшую стрелочку, позволяющую свернуть их и сделать дорожки уже. Таким образом у вас будет больше места и на рабочую область поместится больше дорожек. Для еще более рациональной работы вы можете задать каждой дорожке свое имя, дважды щелкнув по надписям Дорожка 1, Дорожка 2 и так далее. Это особенно важно, когда у вас много дорожек и вам нельзя запутываться.

Вы также можете менять длительность различных фрагментов, удаляя или уменьшая паузы с тишиной (имеется в виду, что сами фрагменты мы не удаляем), или наоборот – удлиняя паузы. В первом случае, выделите участок, который вы хотите удалить и нажмите Delete или Backspace. Чтобы вставить тишину, поместите указатель выделения в нужное место и перейдите в Создание > Тишина, но на этот раз не забудьте указать длительность вставки. Не перестарайтесь, так как, к примеру, три секунды тишины в действительности значат очень много.

Мы выполним еще одно действие: перенесем звуковой фрагмент на отдельную дорожку, после чего его можно будет с удобством двигать относительно других частей записи. Выделите нужный фрагмент и копируйте его в буфер обмена (Правка > Копировать или Ctrl+C). Перед тем как снять выделение, не забудьте заполнить его тишиной, используя описанный выше способ, а затем щелкнете по левой границе выделения, при этом курсор переместится в начало фрагмента. Теперь выберите Проект > Создать новую стереодорожку (Project > New Stereo Track) и вставьте в нее скопированный ранее фрагмент (Правка > Вставить или Ctrl+V).

«В ВАШЕЙ КОМПОЗИЦИИ МОГУТ БЫТЬ ПАУЗЫ, КОГДА ВЫ ЗАХОТИТЕ УВЕЛИЧИТЬ ГРОМКОСТЬ ФОНОВОЙ МУЗЫКИ.»



Звук распределен и сдвинут по дорожкам. Добавлено музыкальное сопровождение. Мы готовы к выходу в эфир...

Теперь у нас есть возможность изменить громкость фрагмента, его стереоэффект, а также его положение на временной шкале, не затрагивая остальную запись. Я в данном случае поместил голос городского головы Гамельна в левый канал (примерно 40% на движке каналов), а самого дудочника – в правый. Таким образом, при воспроизведении эти два голоса будут различаться.

Несмотря на то, что в *Audacity* существует еще множество инструментов редактирования и эффектов, для нашего подкаста пока достаточно проделанной работы. Разнообразные эффекты доступны в одноименном меню, и вы можете поиграть с ними!

Иллюстрируем запись

Теперь, когда весь звук подчищен и распределен по каналам и дорожкам, пора дополнить проект дополнительными частями. Я доволен структурой получившегося произведения, но ему не хватает музыки. У меня есть собственная версия Адажио для струнных Барбера (к сожалению, запись охраняется законом об авторских правах, поэтому использовать ее можно только в нашей статье), которая создаст неплохой волнующий аккомпанемент. Выберите пункт Проект > Импортировать звуковой файл... (Project > Import Audio File) и найдите необходимую запись (для получения информации о музыкальных записях Open Source смотрите также врезку «Вещание и авторское право» на этой странице). Новая запись будет помещена на отдельную дорожку, и вы сможете подрезать ее, чтобы аккомпанемент по длине соответствовал всей композиции.

При нажатии на кнопку Play начнется воспроизведение всех дорожек одновременно, но я хочу, чтобы запись начиналась с музыкального фрагмента, а не сразу с текста. Поскольку дорожки с голосом изначально помещены первыми, нам пригодится инструмент сдвига по времени, который позволит переместить эти записи вправо. Вы помните, что на голос у нас отведены две дорожки, которые нельзя сдвигать относительно друг друга. Поэтому мы выделяем обе дорожки с помощью инструмента выделения и клавиши Shift, а затем перемещаем их инструментом сдвига на 15 секунд вперед (две дорожки перемещаются одновременно). Теперь при воспроизведении сначала играет Адажио, и через некоторое время начинается голос.

Вы можете заметить, что вся запись проигрывается на одинаковой громкости. Для нескольких секунд это допустимо, но слушать так весь текст очень раздражает. Решением этой проблемы является эффект приглушения. Суть его, в двух словах, такова: когда начинается голосовая часть записи, громкость музыкальной дорожки должна быть ниже. *Audacity* позволяет с легкостью создать подобный эффект.

ВЕЩАНИЕ И АВТОРСКОЕ ПРАВО

Авторское право – очень скользкая вещь, и ограничения на использование материала (под которым я понимаю любой авторский оригинал – роман, симфония, картина или что-то еще) меняются очень часто.

Идея авторского права заключается в следующем: после публикации своего произведения автор в течение некоторого времени обладает на него исключительным правом, после чего это произведение становится общественным достоянием. Под «достоянием» понимается право каждого использовать это произведение без разрешения или выплат автору. Длительность периода защиты авторского права стала расти по мере принятия соответствующих законов и на сегодняшний день составляет (в большинстве стран) 70 лет с момента смерти автора. Роберт Браунинг, чей текст я использую в данной статье, умер в 1889 году, поэтому его произведения не защищены авторским правом в наше время.

Право на произведение, приобретшее статус общественного достояния, не может быть заново присвоено кем-либо. Тем не менее, в некоторых случаях происходит конфуз. Если речь идет о тексте – книге или поэме, переведенном с другого языка, то вам следует убедиться, что перевод также не является объектом авторского права. В свою очередь, музыкальный фрагмент может быть общедоступен, но сама запись – подпадать под действие стандартных ограничений.

Начинающим подкастерам будет нелегко с этим разобраться, но, к счастью, существует книга Ларри Лессига (Larry Lessig) «Свободная культура», которая отлично подойдет для начала. Это свободное (во всех смыслах) издание доступно на сайте www.free-culture.cc

При создании подкаста убедитесь, что вы не нарушаете ничьи музыкальные права, в противном случае вас могут взять за шкирку за преступление закона. В качестве альтернативы, вы можете сами сочинить свою музыку в программах типа Rosegarden или LMMS. Для тех, кто не слишком одарен музыкальным чутьем и слухом, существуют тысячи готовых звуков на таких сайтах, как Freesound (http://freesound.iua.upf.edu/searchtext.php), проект Creative Commons (http://creativecommons.org/audio) и Open Directory (http://dmoz.org/Arts/Music/Sound_Files/Samples_and_Loops). Звуки доступны для скачивания либо за разовую плату (без роялти), либо на условиях различных лицензий CC (Creative Commons). Вне зависимости от того, откуда вы достали аудиофайл, вы должны прочесть и принять условия лицензии на него. Это может, к примеру, означать, что в случае с некоторыми лицензиями CC вы обязаны опубликовать работу, где использован этот исходный файл, по такой же лицензии, или вы имеете право использовать данный звук только в некоммерческих целях.

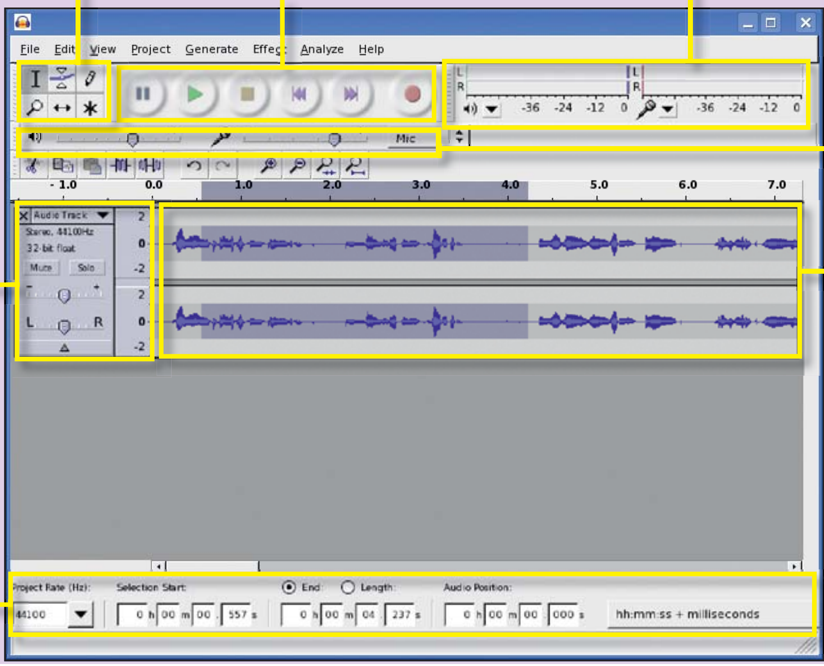
В любом случае, вам действительно стоит ознакомиться с проектом Creative Commons и попробовать подобрать себе удобную лицензию.

ИНТЕРФЕЙС AUDACITY

В этой небольшой панели размещаются шесть инструментов, но нас интересуют только три: Выделение (Select), Смещение (Shift) и Изменение огибающей (Envelope). Первый необходим для всех операций редактирования и позволяет создавать выделения в звуковой дорожке, которые можно копировать, удалять или использовать для применения эффектов.

Панель управления (transport controls)
Такие же кнопки есть на большинстве бытовых проигрывателей. Кнопки перехода к началу и к концу записи работают соответственно самой продолжительной дорожке в проекте. Для перехода к началу или концу выделения используйте клавиши управления курсором на клавиатуре. Если при нажатии на кнопку Play вы будете удерживать Shift, то выделенный фрагмент будет проигрываться «по кругу» – то есть бесконечно, пока вы не нажмете на Stop.

Этот раздел показывает уровни входящего и исходящего сигнала. По умолчанию данный индикатор отключен. Чтобы включить его, щелкнете по маленькой стрелке рядом с иконкой микрофона и выберите «Начать мониторинг» (Start monitoring). Для того, чтобы снизить нагрузку на процессор, установите интервал обновления датчика на большее значение (в разделе настроек), однако если ваш компьютер работает медленно, лучше не включать этот индикатор вообще.



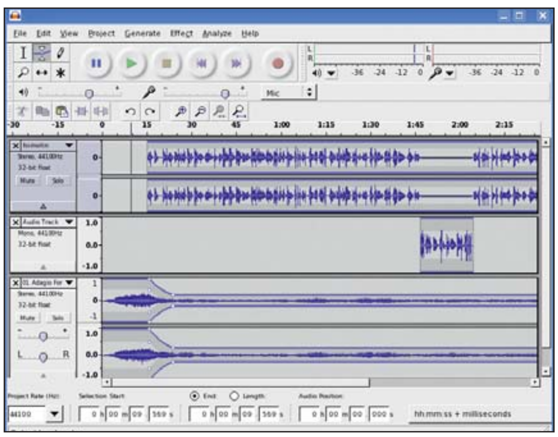
Здесь вы можете настроить каждую дорожку в отдельности: например, отрегулировать громкость. Здесь также есть две полезные кнопки: первая из них – Тишина (Mute) – приглушает звук только в текущей дорожке (все остальные продолжают звучать), а кнопка Соло (Solo) действует наоборот – отключает все дорожки, кроме текущей.

Необязательно делать выделения мышкой. Для более точных результатов предусмотрена панель с числовыми данными.

Эта панель представляет собой урезанную версию обычного регулятора громкости и влияет на уровень входного/выходного сигнала.

Пространство звуковых дорожек служит наглядным графическим отображением звука, и именно здесь происходит основная часть работы. При выбранном инструменте Масштабирование (Zoom) вы можете приближать или удалять звуковые фрагменты при нажатии на левую или правую кнопки мыши соответственно. Полоса горизонтальной прокрутки позволяет перемещаться по самой записи.

Выделите нужную дорожку и щелкните по инструменту Изменение огибающей. Теперь при щелчке левой кнопкой мыши по дорожке будут появляться узловые точки, определяющие профиль приглушения или нарастания звука. Чем ближе положение точек к центральной оси дорожки, тем тише будет звук в этой области. В вашей композиции могут быть паузы или перерывы, когда можно будет немного увеличить громкость звука (это особенно справедливо для выпусков новостей, где музыкальные фрагменты служат разделителями между блоками информации), поэтому вы можете создать дополнительные узлы и регулировать громкость звука в нужных местах.



Перед вами три дорожки: диалог (начитанный мною текст про дудочника) и музыкальное сопровождение (отрывок из Адажио), которое при помощи эффекта огибающей становится тише во время диалога. Подкастика!

Покажите свое творение

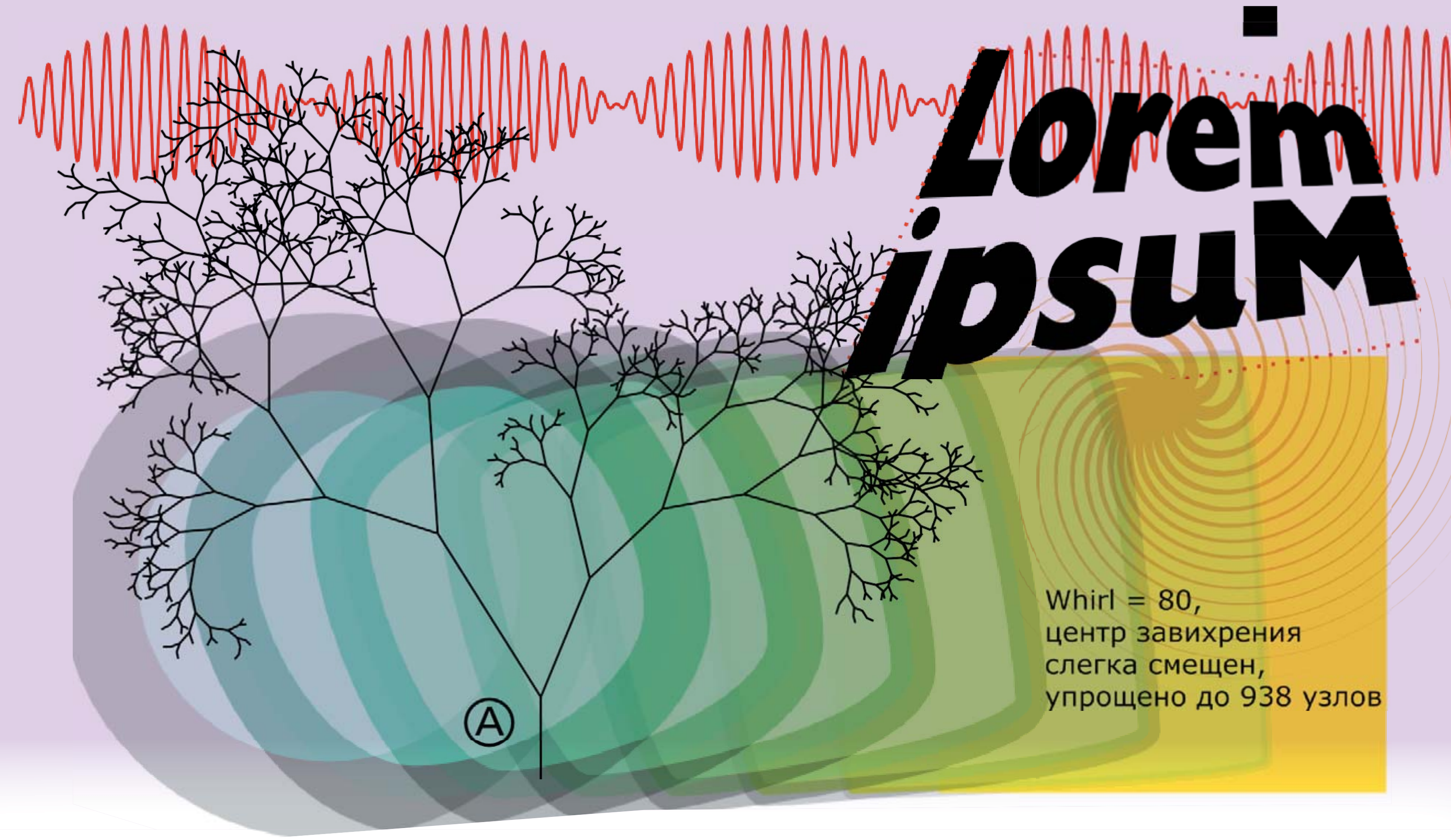
Мы находимся на финальной стадии, которая среди звукорежиссеров называется окончательным сведением. На этом этапе все дорожки склеиваются в одни стерео-файл, который уже можно распространять. Если нам нужно записать звук на CD, то мы экспортируем наш проект как WAV-файл, но в моем случае речь идет о подкасте, который будет распространяться через Интернет и поэтому должен иметь небольшой размер. Вместо WAV мы выбираем для экспорта формат MP3.

Появится предупреждение о сведении всех дорожек в один двухканальный файл – просто нажимаем OK и задаем файлу имя и расположение. По умолчанию, модуль *Lame* выдает качество 128 kbps (килобит в секунду), что является хорошим компромиссом между качеством и размером файла. Тем не менее, если ваша запись состоит целиком из начитанного текста, то вы можете смело сократить битрейт до 64 kbps – это будет не очень страшно, а пользователи коммутируемых соединений скажут вам спасибо. Нужные настройки находятся в пункте меню Правка > Настроить на вкладке Форматы. В разделе настройки экспорта MP3 вы найдете выпадающий список битрейтов, где можно задать нужное значение. Запомните, что меньшие значения означают более скромный размер файла, а большие – лучшее качество.

Экспортированный файл готов к публикации в Интернете, где его смогут услышать миллионы людей. Это ваша потенциальная аудитория. LXF

ЧЕРЕЗ МЕСЯЦ

Мы применим эти навыки в другом проекте и будем использовать *Audacity* для создания саундтрека к вашим домашним видеозаписям.

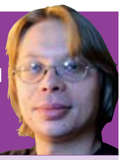


Inkscape Эффектная игра

НА ДИСКЕ

ЧАСТЬ 6 На последнем уроке **Дмитрий Кирсанов** расскажет о расширениях Inkscape – дополнительных командах, создающих подчас действительно необычные эффекты.

МЕСЯЦ НАЗАД



Мы ознакомились с инструментами *Inkscape* для работы с текстом.

Любая программа становится успешной и значимой только тогда, когда она из отдельного приложения вырастает в полноценную платформу. Никакая программа не может удовлетворить всех. Но если любой программист средней квалификации может без труда расширить её возможности и если эти расширения органично вписываются в интерфейс программы – число её потенциальных пользователей резко возрастает.

Примеры таких приложений, ставших платформами, хорошо известны – это ранний *Netscape* с его JavaScript и подключаемыми модулями, нынешний *Firefox* с его бесчисленными расширениями или же *Adobe Photoshop*, чьи расширения образуют целый огромный мир. *Inkscape* сделал только первые шаги в этом направлении, но, похоже, намеревается стать настоящей платформой в области векторной графики и SVG. Эффекты (расширения) *Inkscape*, пусть пока и немногочисленные, уже вполне заслуживают вашего внимания.

Команды, собранные в меню «**Эффекты**» (**Effects**), вызывают сценарии (скрипты), написанные на языках Python (чаще всего), Perl или Ruby. Каждый такой сценарий берет SVG-файл на входе и выдает другой SVG-файл на выходе. Inkscape передаёт сценарию тот документ, который вы в данный момент редактируете, а вместе с ним – параметры командной строки, специфические для каждого эффекта. Значения этих параметров пользователь устанавливает в специальном диалоговом окне перед запуском эффекта.

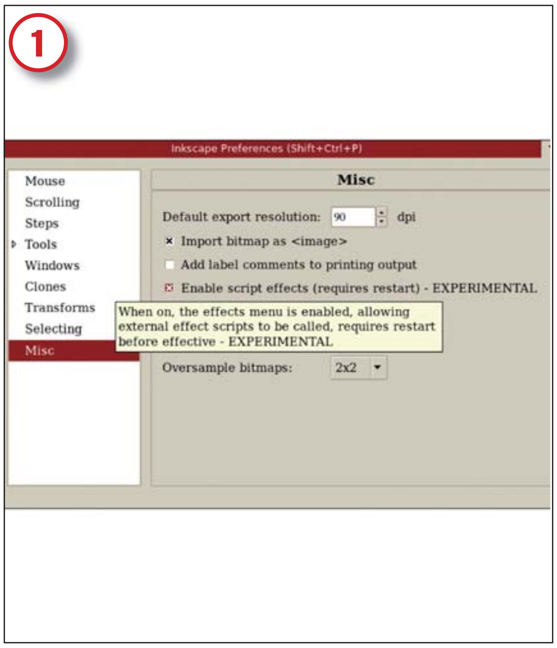
Сценарии эффектов должны быть самодостаточными – после запуска *Inkscape* не оказывает им никакой поддержки. Эффекты не имеют доступа ни к памяти *Inkscape*, ни к его внутренним функциям. (Расширения другого типа, глубже интегрированные с *Inkscape*, сейчас находится в стадии разработки).

Очевидный недостаток такого подхода состоит в том, что сценарий должен выполнять всю работу самостоятельно – читать SVG-файл, отыскивать в нем нужные элементы, изменять их и записывать измененный документ обратно в файл. Пользоваться при этом он может только стандартными библиотеками того языка, на котором написан. Но у этого недостатка есть и оборотная сторона – разработчику не придется изучать внутренности *Inkscape*, чтобы создать эффект. Превратить любую программу для обработки SVG в Inkscape-эффект совсем нетрудно. И наоборот – каждый такой эффект можно использовать как самостоятельную программу.

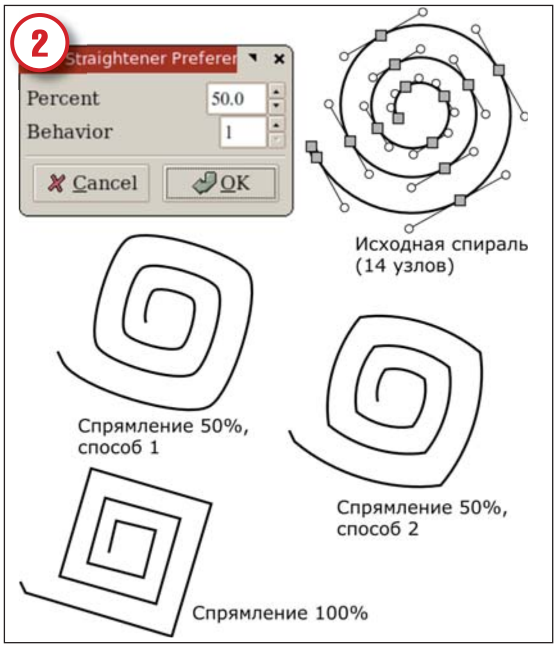
Возможность запускать сценарий прямо из редактора, сразу же видеть результаты его работы, при необходимости отменять изменения и запускать его снова с другими параметрами – все это делает эффекты весьма удобными. Большинство эффектов, включенных в последнюю версию *Inkscape* (0.43), написаны на Python и работают с контурами. Некоторые из них предназначены для узкоспециальных, редко возникающих нужд и, похоже, навсегда обречены быть именно расширениями. Другие, напротив, реализуют базовые функции, которые пока отсутствуют в ядре *Inkscape* – но, вероятно, со временем будут туда добавлены и перестанут быть расширениями.

На этом занятии я покажу вам, как работают некоторые из эффектов и для чего они могут пригодиться.

1



2



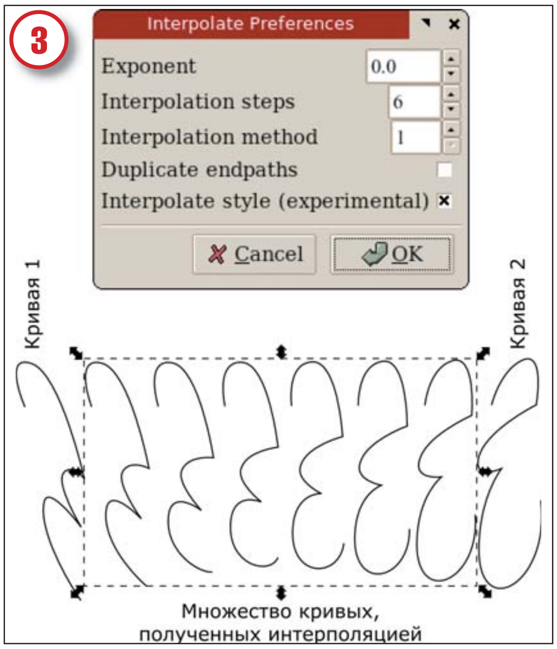
Включение эффектов

В версии *Inkscape 0.43* меню «Эффекты» (Effects) по умолчанию недоступно. Откройте «Настройки Inkscape» (Inkscape Preferences) и на закладке «Прочее» (Misc) установите флажок «Включить эффекты» (Enable Scripts Effects). Перезапустите программу, и вы увидите это меню со списком всех доступных эффектов. В Linux, при установленных Python и PyXML, они работают сразу; наладить эффекты в Windows-версии пока несколько труднее. В версии 0.44 эффекты, по уверениям разработчиков, будут работать сразу и на любой платформе.

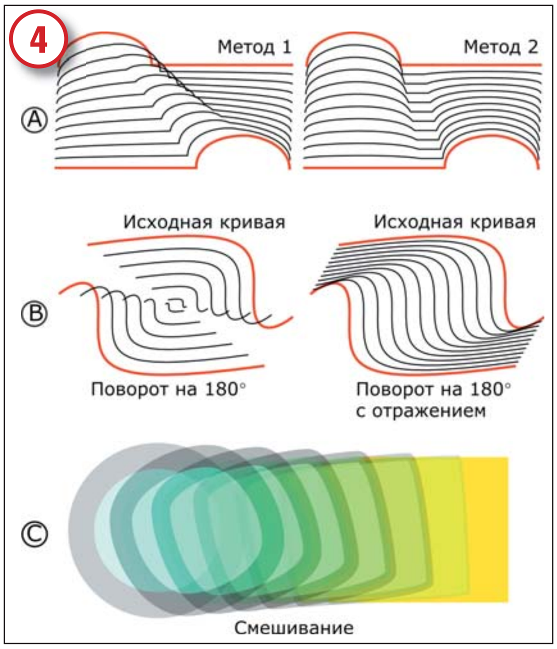
Спрямление сегментов

Попробуем для начала простой эффект, пропорционально укорачивающий все манипуляторы Безье в контуре. Выделите объект (я нарисовал спираль соответствующим инструментом (F9), затем превратил её в контур нажатием Ctrl+Shift+C) и выберите в меню «Эффекты > Спрямитель сегментов» (Effects > Segment Straightener). Можно выбрать как степень, так и алгоритм спрямления: один из них просто укорачивает манипуляторы, а второй укорачивает и поворачивает их к следующему узлу, превращая гладкие узлы в ломаные.

3



4



Эффект интерполяции

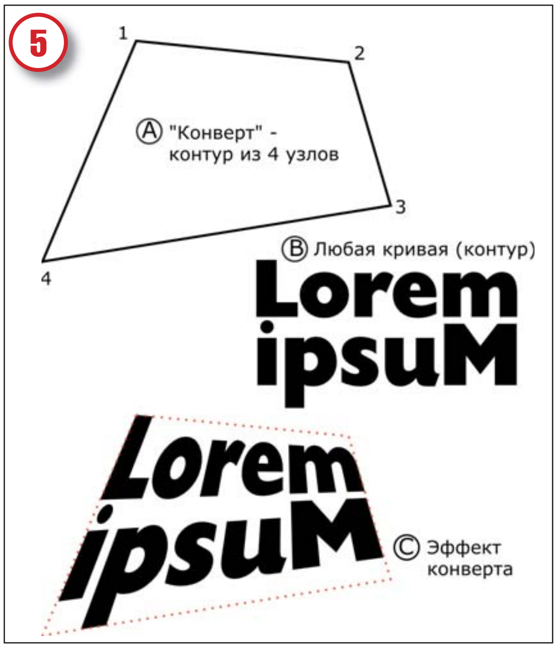
Другой эффект предназначен для создания плавного перехода (интерполяции) между двумя контурами. Нарисуйте пару линий инструментами «Перо» или «Карандаш». Теперь, выбрав «Эффекты > Интерполяция» (Effects > Interpolation), вы можете получить множество промежуточных контуров, постепенно превращающих один объект в другой. В диалоговом окне можно выбрать количество шагов интерполяции и «экспоненту» – число, определяющее равномерность интервалов между ступенями.

Выбор метода

Интерполяция хорошо работает с контурами, которые и без того достаточно похожи и не имеют субконтуров (например, дыр). Метод 1 сохраняет существующие узлы и больше подходит для простых линий. Метод 2 в случае необходимости добавляет новые узлы и хорош для более сложных случаев (A). В некоторых случаях одну из линий приходится реверсировать (B). Опция смешивания стилей – цветов заливки, толщины линий, прозрачности – хоть и заявлена как экспериментальная, но работает вполне удовлетворительно (C).

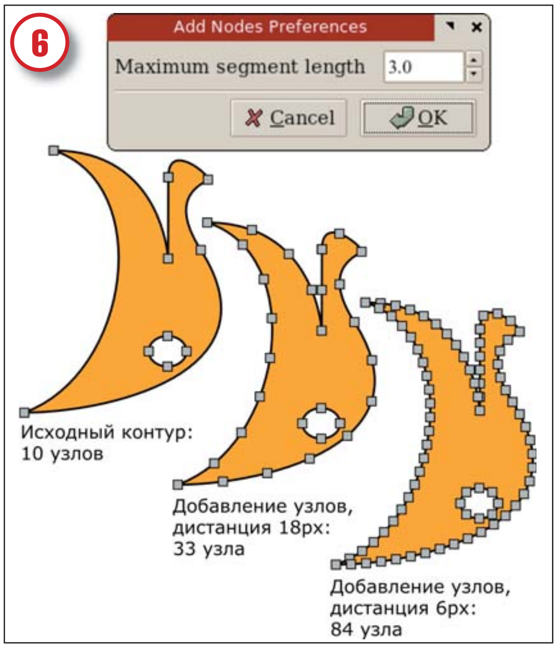
ПОДСКАЗКА

- Некоторые эффекты вставляют создаваемый объект в корень документа вместо текущего слоя. Проблему можно обойти так: вырежьте объект (Ctrl+X) из корня, переключитесь в нужный вам слой и вставьте объект на его место сочетанием клавиш Ctrl+Alt+V.
- Наряду с эффектами, *Inkscape* поддерживает расширения ввода/вывода для работы с дополнительными форматами импорта и экспорта (например, для импорта PostScript). Обычно для работы таких расширений требуется установка вспомогательных приложений.
- По команде «Создать изображение» (Create Bitmap) Inkscape может пропустить создаваемое изображение через внешний фильтр. Например, в состав дистрибутива *Inkscape* входит сценарий, позволяющий быстро получить размытые тени (для этого используется ImageMagick; подробнее об этом можно прочитать в файле **doc/inkscape-shadow.README**).
- В версии 0.44 добавлен эффект-сценарий для встраивания растровых изображений в SVG-документ. Дело в том, что по умолчанию импортирование растровой картинки вставляет в документ лишь ссылку на нее. А после встраивания SVG-файл хранит в себе и векторную, и растровую части, – так что его гораздо удобнее, например, посылать по почте. Другой сценарий позволяет вынуть встроенный растр из SVG и записать его в отдельный файл.



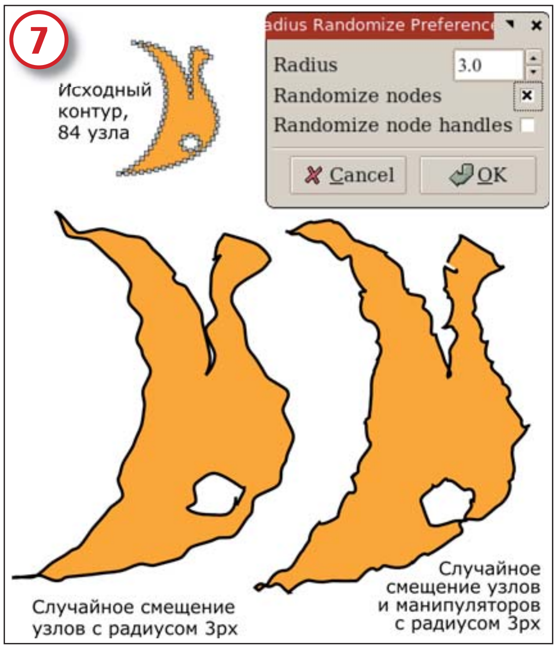
Конверт

Другой эффект, который, надеюсь, со временем станет частью ядра *Inkscape* – «Конверт» (Envelope). Чтобы воспользоваться им, нужно выделить два контура, причём один из них должен иметь ровно четыре узла (A) – он послужит конвертом для вставки второго (B). К сожалению, пока этот эффект не вносит перспективных искажений и не может использовать криволинейные конверты. Чтобы восполнить пробел, Inkscape нужна помощь талантливых программистов!



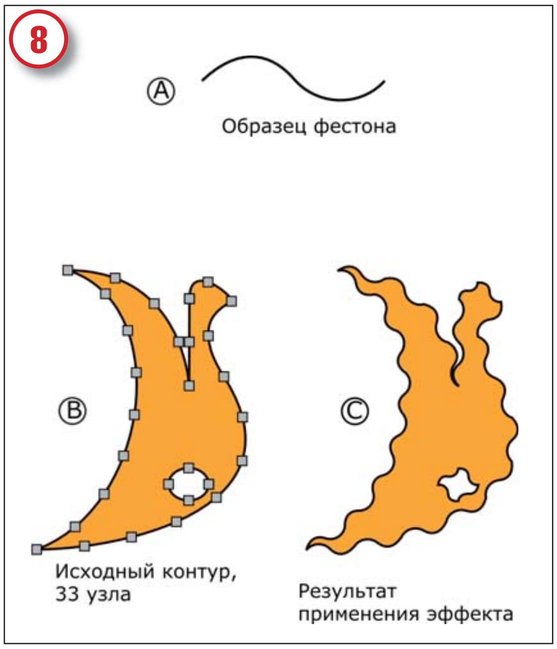
Добавление узлов

Как правило, чем больше узлов имеет контур, тем естественнее выглядит результат работы эффекта. Конечно, можно добавлять узлы вручную при помощи инструмента редактирования узлов (F2), но есть и более удобное средство: эффект «Добавление узлов» (Add Nodes). Вы можете задать максимальное расстояние (в единицах SVG) между узлами по контуру, и эффект создаст столько узлов, сколько необходимо для соблюдения этого расстояния, распределив их по возможности равномерно. Форма контура при этом не меняется.



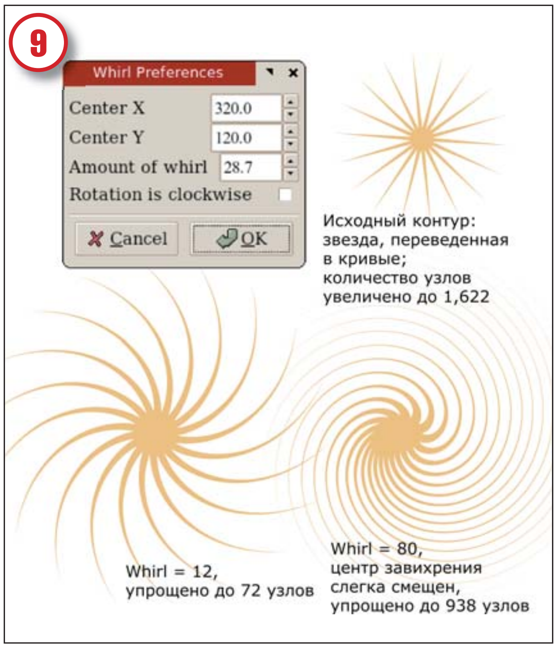
Рандомизация

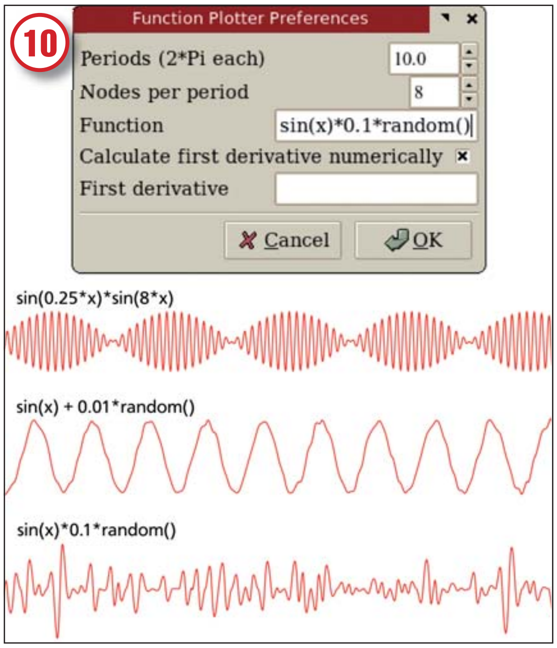
После добавления узлов попробуем «растрясти» контур эффектом «Случайное смещение» (Radius Randomize). Задайте радиус (максимальное смещение) и выберите, к чему вы хотите применить смещение – только к узлам, к их рукояткам или к тем и другим одновременно. Применение только к узлам сохраняет гладкость узлов (A), тогда как смещение рукояток делает контур зазубренным (B). Вручную меняя плотность узлов на разных участках контура, можно регулировать степень его «огрубления».



Фестоны

Эффект «Фестоны» (Kochify) заменяет каждый отрезок контура (между двумя узлами) соответствующим образом масштабированным и развёрнутым образцом. Понятно, что и в этом случае на результат эффекта прямо влияет количество узлов в контуре: чем чаще идут узлы, тем мельче будут вставленные между ними образцы. Прежде всего, необходимо выбрать образец командой «Эффекты > Фестоны (Загрузить)» (Effects > Kochify (Load)) (A). Затем выделите контур, к которому будет применён эффект (B), и выберите «Эффекты > Фестоны» (Effects > Kochify) (C).



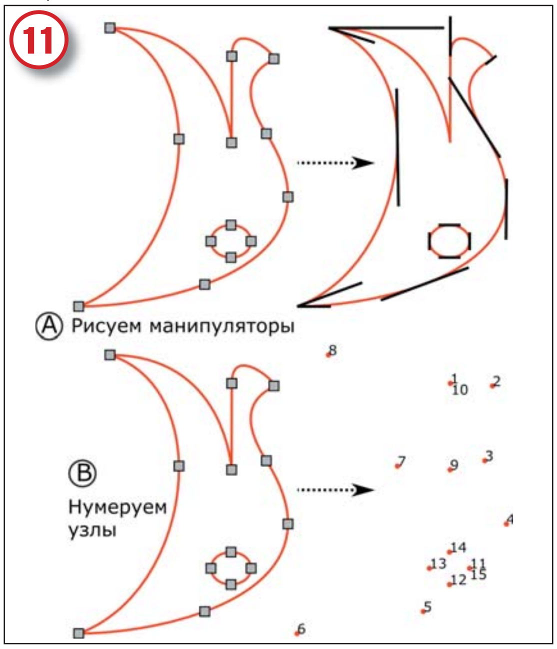


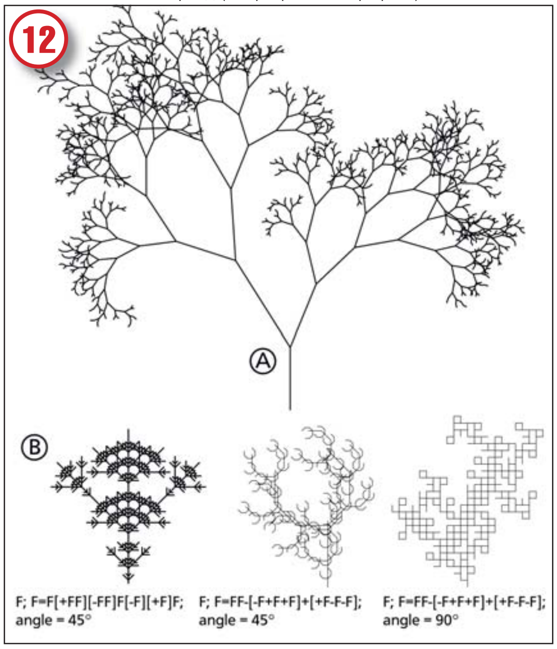
Завихрения

Эффект «Вихрь» (Whirl) закручивает все узлы контура вокруг некоего центра, увеличивая угол отклонения пропорционально радиусу. Этот эффект лучше всего смотрится с центрально-симметричными фигурами, такими как звёзды, и становится тем выразительнее, чем больше узлов вы добавите (закончив работу, лишние узлы можно убрать командой «Упростить», Simplify). Координаты центра вращения нужно указывать численно (0,0 соответствует левому верхнему углу), что немного неудобно, но зато даёт возможность поиграть с эксцентрическими завихрениями.

Графики функций

Эффект «График функции» (Function Plotter) позволяет нарисовать график любой математической функции. Прежде чем вызывать его, выделите прямоугольник, в который будет вписан график. Формулы записываются по правилам Python и могут содержать вызовы любых библиотечных функций из модулей Math или Random. Первая производная (она нужна для определения угла манипуляторов) может быть задана аналитически пользователем или вычисляться самой программой. Можно также указать количество отображаемых 2π-периодов и количество точек на период (т.е. разрешение графика).





Визуализация контуров

Иногда могут быть полезны эффекты «Показать манипуляторы» (Draw Handles) и «Цепочка точек» (Connect The Dots). Первый из них создает новый контур, соответствующий рукояткам всех узлов выделенного контура (A). Второй эффект заменяет контур пронумерованной последовательностью круглых точек, по одной на узел (размеры этих точек и шрифт номеров можно менять). Заметьте, что в замкнутом контуре начальная/конечная точка будет иметь двойную нумерацию – так, в (B) номер 1 накладывается на 10, а 11 на 15.

Рекурсивные фигуры

Последние два эффекта, поставляемые с *Inkscape*, создают красивые фракталы при помощи рекурсивных алгоритмов. Первый, «Случайное дерево» (Random Tree) (A) рисует ветвистое дерево со случайными углами наклона и длиной ветвей; параметр «Минимальный размер» (Minimum size) относится к размеру веток и определяет глубину ветвления. Другой эффект, Lindenmayer (B), не использует элемент случайности, но позволяет точно управлять движением пера на каждом шаге рекурсии при помощи L-системы (подробнее об этом см. http://en.wikipedia.org/wiki/L-system).

НАПОСЛЕДОК...

Закончено последнее занятие этой серии. Возможно, скоро мы встретимся снова, а пока – играйте с *Inkscape*, пробуйте этот новый инструмент в реальных проектах. Комментарии к настоящей серии и идеи для будущей принимаются по адресу: letters@linuxformat.ru.



РЕДАКТИРОВАНИЕ ЗВУКА

Аудио Чистим записи от шума

Наши драгоценные записи могут оказаться слишком уязвимы. Неважно, пожевал ли кот пленку, или вы просто пролили пиво на пульт, **Грэм Моррисон (Graham Morrison)** поможет очистить звук.

НА ДИСКЕ

НЕОБХОДИМЫЕ ИНСТРУМЕНТЫ

- *Audacity*
- *Ardour*
- Jack
- LADPSA plugins

Желательны также:

- *Qjackctl*
- *Jaaa*
- *Postfish*

Шум в аудиозаписи может возникнуть по разным причинам. Возможно, при записи на виниловую пластинку попало слишком много «атмосферы», или чувствительность микрофона была слишком низка для того, чтобы звук преобладал над фоновыми шумами. Ну и конечно, у всех нас есть кипы аудиокассет, которые мы надеемся сохранить в XXI веке. Используя набор специальных Linux-инструментов и приложений, у вас есть возможность вдохнуть новую жизнь в эти старые записи и сделать их звучание таким же, как и в момент создания. Существует три типа шумов, которые портят звук: широкополосный шум, постоянный фоновый шум, а также импульсные помехи – щелчки, треск и другие дефекты.

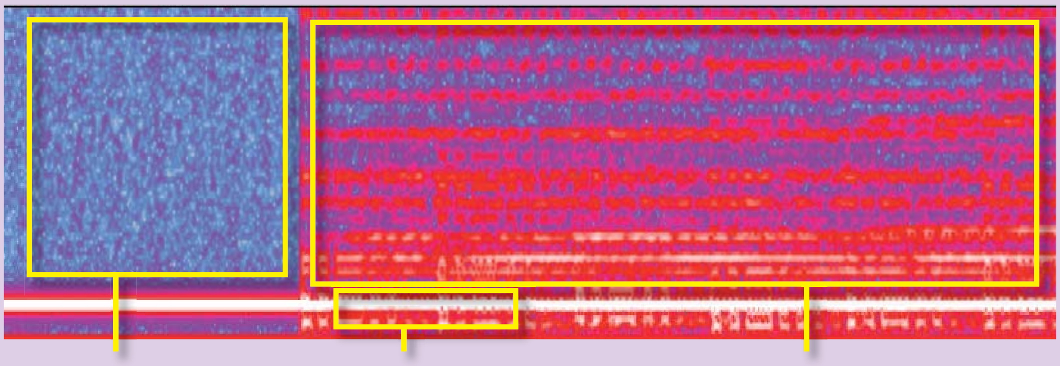
Искусство шума

Широкополосный шум получил свое имя из-за того, что он присутствует на всем спектре частот. Больше всего он заметен на высоких частотах, и вы услышите его практически на любой аналоговой записи. Это так называемый «белый» шум – звук ненастроенного телевизора. Каждое устройство, через которое проходит ваш аудиосигнал – включая микрофон и аналого-цифровой преобразователь на звуковой карте – добавляет некоторое количество такого шума. Поэтому чем выше качество вашей техники, тем меньше широкополосного шума будет создано при записи.

На втором месте находится нежелательный фоновый шум/гул на заднем плане. Подобно широкополосному шуму, он порождается оборудованием, но при этом заполняет вполне определенный спектр частот и на слух различается как низкий тон. Такой шум является результатом заземления электрооборудования при котором создается «петля» - слабые токи, которые проходят по аудио-кабелям. Частота колебаний такого шума соответствует частоте переменного тока, но это не опасно ни для человека, ни для звука. Промышленная частота тока составляет 50Гц в Великобритании и России и 60Гц в США, что воспринимается на слух как очень низкий бас. Вы также можете расслышать высшие гармоники этих колебаний, которые будут иметь частоты из ряда 100Гц, 200Гц и так далее. Еще одним источником гула служит ваша звуковая карта, которая имеет свойство ретранслировать высокочастотные наводки остальных компонентов компьютера. Из-за этого производители специально экранируют качественные звуковые карты.

Импульсные помехи бывают как аналоговыми, так и цифровыми. Первые чаще всего встречаются при перезаписи старых виниловых пластинок, когда игла натыкается на загрязненный участок дорожки и подпрыгивает (треск). Цифровые щелчки являются либо результатом ошибки кэширования (процессор не успевает за ходом записи), либо возникают из-за чрезмерной громкости сигнала, когда тот зашкаливает за границы диапазона и часть звука попросту «отрезается».

АНАТОМИЯ ШУМА И СПЕКТРАЛЬНЫЙ АНАЛИЗ



Широкополосный шум. Он присутствует «на заднем плане» на протяжении всей записи.

Фоновый гул и шипение занимают определенный набор частот и также заметны на всех участках записи.

Вот так выглядит ваша музыка. Рисунок постоянно меняется, но в нем постоянно присутствуют перечисленные выше разновидности шума.

Самый простой способ управиться с низкокачественной записью состоит в использовании визуальных средств. С их помощью вы сможете определить характер шума и наиболее эффективно удалить его. К сожалению, даже несмотря на свои научные корни, набор инструментов визуализации в Linux совсем невелик. Классическим способом представления звука является периодограмма, используемая практически любым звуковым редактором. По вертикальной оси фиксируется амплитуда сигнала, по горизонтальной – время, а результатом является знакомый рисунок из «пиков» и «впадин». Волновой график прекрасно подходит для обнаружения щелчков и других резких изменений амплитуды, но он бесполезен при работе с низким шумом и гулом. Для них требуется что-то наподобие спектрограммы, где амплитуда каждого набора частот выделена своим цветом (см. врезку «Анатомия шума» вверху страницы). По горизонтальной оси отложено время, по вертикальной – частота. Синим цветом отмечен сигнал с небольшой, красным – со средней, и белым – с высокой амплитудой.

Audacity имеет ряд инструментов для анализа звука и входит в состав многих дистрибутивов как стандартный аудиоредактор. В нем возможно переключение между режимами графика частот и спектрограммой, а также есть окно анализа выделенного звукового фрагмента. Для того, чтобы воспользоваться другими инструментами, вам понадобится протокол реального времени Jack, который является единственным способом применять звуковые эффекты и сразу же слышать результат.

ШАГ 1 – УДАЛЯЕМ ИМПУЛЬСНЫЕ ПОМЕХИ

Щелчки, также известные как «обрезки» (clipping), представляют собой наиболее пагубную разновидность шума, так как они перекрывают полезный сигнал, не оставляя от него следов. Обрезанный сигнал раздражает хуже нарыва, так как звук переходит от нормальной громкости к очень высокой и зашкаливает в течение нескольких выборок [цифровой звук имеет дискретную природу, поэтому при оцифровке производится определенное количество выборок – замеров – в единицу времени, – прим. ред.] Если под обрез попал большой участок, то вы, скорее всего, ничего с этим не сделаете. Тем не менее, более мелкие помехи можно исправить вручную.

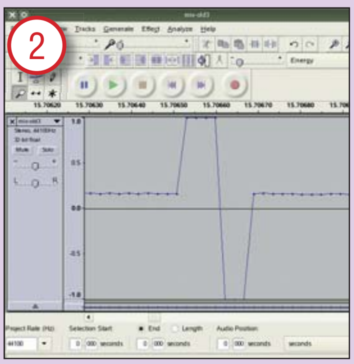
Чтобы сгладить «рваные» участки записи, нам нужно отредактировать исходный сигнал. В стандартном CD-качестве одна секунда звука состоит из 44100 выборок, каждая из которых имеет 16-разрядную амплитуду. *Audacity* позволяет прекрасно работать на уровне этих сигналов: просто загрузите файл и, используя лупу, увеличьте какой-либо фрагмент до максимума – пока не увидите отдельные сигналы. Небольшой щелчок (около дюжины выборок) можно «срисовать» с ближайшего здорового участка. Не забывайте почаще проигрывать обрабатываемый кусочек записи. Если искажения заметны на слух, это означает, что вы неправильно скорректировали профиль звуковой волны.

АНТИШУМ: ЩЕЛЧКИ



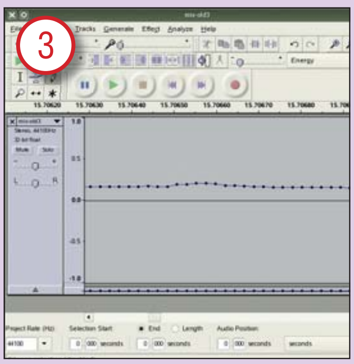
Идентифицируем

Найдите щелчки в окне *Audacity*. Они выглядят как резкие скачки амплитуды.



Увеличиваем

С помощью лупы увеличьте масштаб настолько, чтобы рассмотреть отдельные сигналы.



Зачищаем

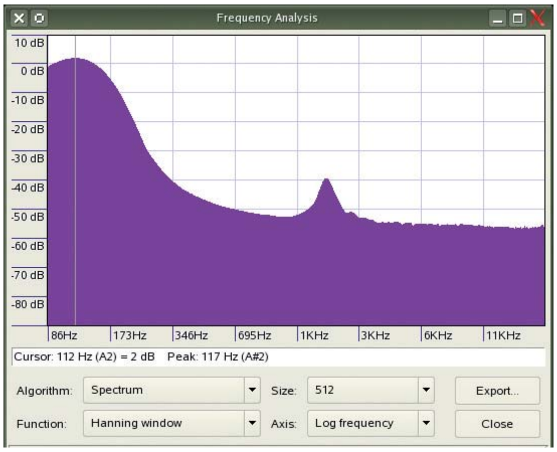
Инструментом Карандаш (Pencil) восстанавливаем линию на уровне соседних, «здоровых» участков записи.

ШАГ 2 – УБИРАЕМ ФОНОВЫЙ ШУМ И ГУЛ

Лучший способ избавиться от гула, жужжания или других тонов известной частоты заключается в использовании эквалайзера. Вы можете увеличить или уменьшить амплитуду определенного набора частот, поэтому эквалайзер – почти идеальный инструмент удаления нежелательных звуков. Минусом можно считать некоторую неразборчивость эквалайзера, который вместе с шумом удаляет и часть полезного звука.

Самое трудное – определить диапазон частот, который соответствует нежелательным звукам и шуму. Вы можете положиться на свой слух и вручную подстроить частоты, но те из нас, кто не одарен идеальным слухом, захотят воспользоваться наглядным инструментом. Просто щелкните по опции Plot Spectrum в *Audacity,* и перед вами возникнет сечение выбранного фрагмента. Низкие частоты отображаются в левой части графика, а высокие – в правой. Кривая отражает относительную амплитуду для разных частот. Постоянный фоновый шум наглядно виден здесь как пик, одинаковый по всей длине записи. К сожалению, *Audacity* не умеет обновлять график в реальном времени – по мере проигрывания звука. Для того, чтобы воспользоваться этой удобной функцией, нам понадобятся *Ardour* и Jack.

Сперва запустите Jack, воспользовавшись утилитой *Qjackctl*, затем *Ardour*. Чтобы сразу видеть все изменения на экране, запустите *Jack and Alsa Audio Analyser*, который скрывается за страшным словом *Jaaa*. В общих словах, это визуальный анализатор спектра, очень похожий на встроенный в *Audacity*. Он может работать с аудиофайлами или с потоком Jack. Для поддержки последнего запустите *Jaaa* с соответствующим ключом: **jaaa -J**. Когда все загружено, проверьте монитор соединений Jack и убедитесь, что звуковой поток направлен правильно. Вывод *Ardour*, как и *Jaaa* (если он запущен), должен быть направлен на выход звуковой карты. В *Ardour* вам необходимо создать новую дорожку: щелкните правой кнопкой мыши по рабочей области и выберите из выпадающего списка Stereo. Потом сделайте еще такой же щелчок и выберите Insert External Sound File (Вставить Звуковой Файл), после чего вам нужно будет указать файл, который вы хотите очистить. Возможно, придется сначала конвертировать этот файл в формат WAV, но это зависит от того, что поддерживает ваша сборка *Ardour*.



Этот график показывает «поперечный срез» нашей записи (время идет как бы по оси z – из глубины страницы на нас). Нам нужно вырезать звук на частоте 100Hz.

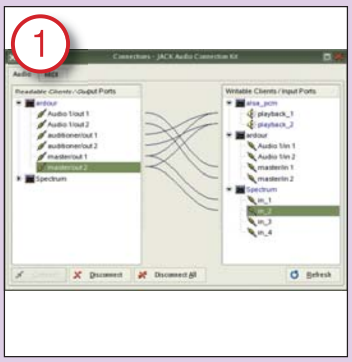
Для загруженного файла сразу же создается амплитудный график. Заметим, что дорожек с файлами может быть сколько угодно (просто для данного примера достаточно одной). После нажатия на кнопку Play все дорожки проигрываются одновременно как один сведенный канал.

Эквалайзер

Для того, чтобы *Ardour* вырезал шум, нужно применить к звуку эффект эквалайзера. В отличие от *Audacity*, в *Ardour* эффекты сразу же применяются к текущей звуковой дорожке. Щелкните на кнопке Editor Mixer (Редактор каналов) в левом верхнем углу окна программы, и вы увидите область точной подстройки каналов. Внизу находится индикатор и ползунок громкости, а сразу же над ними вы заметите квадратик (под кнопкой записи), который и отвечает за эффекты. Щелкните правой кнопкой мыши по нему и выберите Новый модуль (New Plugin). Появится окно со списком всех установленных у вас модулей LADSPA. Здесь вы найдете самые разные фильтры, упорядоченные по типу, но нам больше всего подошел 4-полосный параметрический фильтр (4-Band Parametric Filter). Его параметричность заключается в том, что, указав частоту шума/гула, вы можете уменьшить ее вклад в запись.

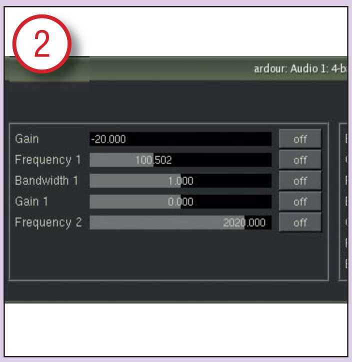
Каждый подключенный эффект отображается на области каналов – вы заметите его по квадратным скобкам, которые означают, что данный эффект загружен, но не активирован. Активация производится в контекстном меню (правый щелчок мыши). Там же присутствует и пункт Edit, отвечающий за подстройку эффекта. В вашем распоряжении четыре полосы, но нам пока достаточно будет всего одной, поэтому для Frequency 1 мы зададим значение, соответствующее частоте обнаруженного нами шума. Теперь просто уменьшите величину присутствия этой частоты в звуке, и шума станет меньше. Эффект можно контролировать на слух, чтобы тут же подобрать оптимальные значения частот, или использовать *Jaaa* для сравнения спектрограммы до и после включения эффекта. Добившись нужного результата, экспортируйте обработанный звук через меню Session.

АНТИШУМ: ГУЛ И ЖУЖЖАНИЕ



Настройка Jack

Для того, чтобы применять эффекты в реальном времени, нам понадобится Jack, а также визуализаторы *Ardour* и *Jaaa.*



Эквалайзер

Загрузите в *Ardour* «шумную» запись и примените к ней эквалайзер. Найдите диапазон частот шума и уменьшите его присутствие в записи.



Проверка на выходе

До и после эквалайзера.
Уменьшите ширину частотной полосы в эквалайзере. Это снизит потери в частотах, близких к гулу.

ШАГ 3 – УДАЛЯЕМ ШИРОКОПОЛОСНЫЙ ШУМ

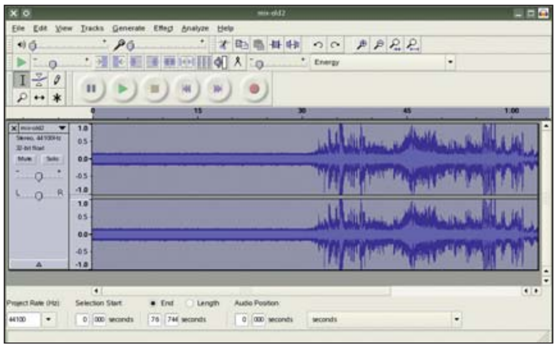
Этот шум труднее всего подается удалению, так как его почти невозможно вычленить из исходного сигнала. Проще всего можно сделать так: заполнить паузы и все тихие участки записи тишиной. Это могут быть промежутки между песнями или задержки в диалогах людей (широкополосный шум наиболее заметен, когда кроме него больше ничего нет). При помощи обычного амплитудного графика в *Audacity* можно определить, когда на фоне шума возникают всплески полезного сигнала. Можно, конечно, удалить шум вручную, изменяя кривую громкости, но есть другой, гораздо более эффективный способ – порог.

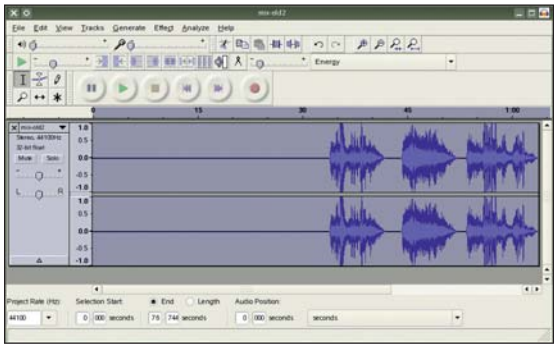
Для начала необходимо выяснить уровень громкости шума. Для этого найдите в *Audacity* фрагмент записи, где нет ничего, кроме шума. Попробуйте проиграть самое начало записи и нажмите паузу до того, как начнется полезный сигнал. Графические индикаторы покажут громкость сигнала в этом месте в децибелах. Запишите это число, так как оно нам понадобится для выставления настроек порога. Среди LADSPA-эффектов содержится множество фильтров, которые действуют по этому принципу, но нам понадобится только один из них, который так и называется – Порог (Gate). Выделите всю запись и перейдите к этому фильтру через меню Эффекты. Бывает что навигация по этому меню превращается в настоящий кошмар, когда сотни установленных эффектов расположены в алфавитном порядке. Появится окно подстройки фильтра, содержащее восемь параметров. Нас интересует только один из них – Threshold (величина порога). Вводим сюда записанное ранее значение и нажимаем кнопку Preview, которая позволит нам убедиться, что из записи действительно удален только шум, а не сам звук. Для наилучшего результата можно слегка поиграть с величиной порога и под конец нажать OK.

Тем не менее, использование порога удалит шум только из пауз и тихих участков, тогда как в составе звука он по прежнему останется. Здесь на помощь приходит более развитый инструмент *Audacity* – шумоподавитель, который вы можете найти в меню эффектов. Этот умный фильтр способен вычленить шум из полезного сигнала и заметно улучшить качество последнего. Но есть и недостаток: в качестве побочного эффекта в записи появляется странный эффект, похожий на звон.

Сначала нужно выделить область, содержащую только шум (начало записи и прочие паузы). Потом нужно открыть этот фильтр и нажать кнопку «Получить профиль шума» (Get Noise Profile). Таким образом у *Audacity* появится образец «чистого» шума, который и будет вычитаться из полезного сигнала. Чтобы применить этот эффект, выделите всю запись и откройте окно фильтра заново, после чего прибавьте действия эффекта, передвинув движок вправо (примерно на 30%) и нажмите OK. После того, как *Audacity* закончит свои вычисления, вы заметите, что запись больше не содержит шума. Качество работы этого фильтра во многом зависит от того, какой образец шума вы выбрали, так что попробуйте несколько вариантов.

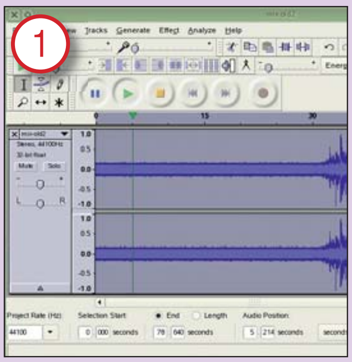
В зависимости от того, с каким рвением вы применяли фильтр, вы можете услышать различные звоноподобные артефакты. Это и есть плата за шум, поскольку вместе с ним из сигнала удаляется и часть полезных частот. Единственный способ с эти справиться – использовать более щадящие настройки фильтра и внимательнее подбирать образцы шума. Частично скрыть этот дефект также поможет эффект реверберации – это небольшое эхо, которое добавляет в звук объема, а также смазывает нежелательные артефакты шумоподавителя. LXF





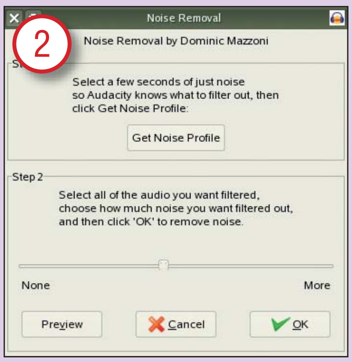
До и после: применив фильтр порога, вы получите тишину в тех местах, где действительно ничего не должно звучать.

АНТИШУМ: ШИРОКОПОЛОСНЫЙ ШУМ



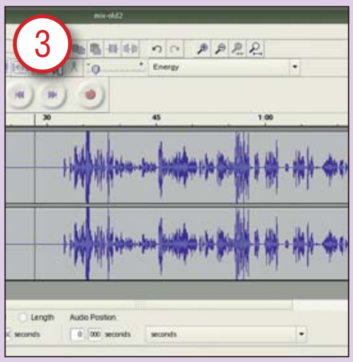
Находим шум

Это фрагмент записи, где нет полезного сигнала – одно шипение и гул.



Создаем профиль шума

Выделив образец шума, перейдите в фильтр Эффекты > Удаление шума и щелкните по кнопке Получить профиль шума.



Обрабатываем запись

Вернитесь обратно к записи и выделите ее полностью. Теперь заново откройте фильтр и нажмите OK.

ВАШ ВЫХОД!

Если у вас есть записи, которые нужно почистить, я надеюсь, что смог вдохновить вас на такой шаг. Кассетная пленка не вечная!

Ваш труд не обязательно будет тяжелым: подстройте эквалайзер, добавьте реверберации – и вполне возможно, что старые записи волшебным образом оживут. Не забудьте поделиться своим опытом и советами с другими меломанами на нашем форуме. www.linuxforum.ru



АВТОМАТИЗАЦИЯ РАБОЧЕГО СТОЛА

DCOP Управляй приложениями KDE

Добавить новые возможности KDE-приложениям и контролировать их удалённо – нет ничего проще, если вы используете DCOP. **Нейл Ботвик** (**Neil Bothwick**) покажет вам, как это делается.

Хотя DCOP даёт вам огромные возможности по управлению вашей системой, вы скорее всего вообще никогда о нём не слышали (разве что, может быть, видели, как это слово мелькало в числе непонятных сообщений, отображающихся при запуске KDE-приложения из окна терминала). Средний пользователь KDE даже не подозревает про Desktop Communication Protocol, хотя он и является фундаментальной частью KDE. Но я думаю, что вам понравится разбираться в нём в процессе чтения этой статьи.

DCOP – это протокол межпроцессного взаимодействия, то есть с его помощью разные KDE-приложения могут общаться друг с другом. Например, именно через него вы можете запустить *KMail* и *KAddressBook* из приложения *Kontact*. Я собираюсь показать вам, как вы можете использовать DCOP для взаимодействия с программами KDE из командной строки или из сценария, добавлять возможности и автоматизировать операции, чего иначе было бы невозможно добиться.

Определение макроса звучит так: это серия нажатий на клавиши, которая эмулирует отсутствующую, но нужную команду. DCOP позволяет добиться того же самого в KDE: добавить функциональность, которая изначально отсутствовала. Чтобы понять, что я имею в виду, посмотрите на следующий пример использования DCOP:

```
rm -i $(dcop kdesktop KBackgroundIface currentWallpaper 0)
```

В моем профиле *Bash* определен псевдоним для этой команды, и я использую его всякий раз, когда мне нужно удалить фоновую картинку на рабочем столе. У меня есть вредная привычка собирать картинки всевозможных сортов и складывать их в тот каталог, из которого KDE берёт обои для рабочего стола, изменяя их через определённый интервал. Очень часто оказывается, что оказавшаяся на рабочем столе картинка мне не нравится... И тогда при помощи этой команды я нахожу файл текущей картинки рабочего стола и удаляю его.

СЦЕНАРИИ ОБОЛОЧКИ

Хотя вы и не обязаны уметь создавать сценарии оболочки для того, чтобы управлять KDE при помощи DCOP, эти знания могут вам помочь. Как правило, сценарий оболочки выглядит примерно так:

```
#!/bin/sh
command1
command2
...
```

Первая строка указывает, что данный файл является сценарием оболочки (в отличие от большого числа других сценариев и исполняемых файлов), а в остальных содержится список команд, которые будут исполняться так, словно вы набрали их в командной строке. После создания текстового файла его нужно сделать исполняемым командой

```
chmod +x yourscript
```

В этом сценарии dcop запрашивает имя текущего изображения на рабочем столе у программы *Kdesktop*. Число 0 обозначает текущий рабочий стол; чтобы сослаться на первый, вы можете использовать число 1, и так далее. Результат выполнения команды *dcop* передаётся команде rm, и ужасная картинка исчезает.

Простой путь к DCOP

Как узнать, что *dcop* может сделать с программой вроде *Kdesktop* и какие аргументы при этом используются? Один из способов – это исследование исходного кода программы и попытка понять, как в ней реализован интерфейс DCOP (это средство, с помощью которого программы обмениваются DCOP-сообщениями). К счастью, есть и более простой способ – KDCOP. Эта графическая программа показывает в виде дерева все приложения, в которых реализованы интерфейсы DCOP. Открыв ветку приложения, можно увидеть список поддерживаемых интерфейсов, каждый из которых включает множество функций. Большую часть интерфейсов вы можете смело проигнорировать, из них вам нужен один единственный, который обычно называется MainApplication-Interface.

Для тестирования функции дважды щелкните по ней. Результатом будет либо возвращенное функцией значение, либо реакция самого приложения. Например, двойной щелчок на Kdesktop > KbackgroundIface > ChangeWallpaper приведёт к тому, что фоновое изображение рабочего стола изменится на следующую картинку из списка (если вы, конечно, указали KDE использовать серию картинок для фона рабочего стола).

Внизу окна KDCOP вы можете увидеть, какую команду DCOP вы только что выполнили. Для ее вызова при помощи утилиты *dcop*, нужно лишь заменить точки на пробелы. Например, следующая функция

```
DCOP call kdesktop.KBackgroundIface.changeWallpaper()
```

может быть вызвана при помощи команды

```
dcop kdesktop KBackgroundIface changeWallpaper
```

Если функция требует какие-то аргументы, перечислите их без скобок через пробел, как это принято в сценариях командной строки.

Другой способ посмотреть доступные операции – утилита командной строки *dcop*. Например,

```
dcop kdesktop KbackgroundIface
```

покажет список функций, доступных в *kdesktop*, в том числе, CurrentWallpaper.

Вы могли заметить, что названия некоторых узлов первого уровня в дереве KDCOP содержат не только имена приложений, но и числа. Это идентификаторы процессов (PID), они добавляются в том случае,

когда одновременно может быть запущено больше одной копии приложения. Например, каждое окно *Konqueror* представляет собой отдельный процесс, так что все вхождения *Konqueror* в KDCOP указываются вместе с PID, даже если в данный момент открыто всего лишь одно окно. С другой стороны, *Amarok* можно запустить только в единственном экземпляре.

Попытка запустить его повторно приведёт к тому, что будет открыто окно текущего приложения. Поэтому ветвь *Amarok* не сопровождается номером.

Для того, чтобы использовать интерфейс DCOP для «нумерованного» приложения, вы обязательно должны правильно указать номер процесса. Существуют разные способы узнать нужный номер из сценария командной строки. Если вы запустили приложение из того же сценария, то PID процесса вам уже известен – он хранится в переменной **$!**.

```
konqueror http://www.site1.com &
dcop konqueror-$! konqueror-mainwindow#1 newTab http://www.site2.net
dcop konqueror-$! konqueror-mainwindow#1 newTab http://www.site3.org
dcop konqueror-$! konqueror-mainwindow#1 newTab http://www.site4.co.uk
```

Первая строка открывает окно *Konqueror* и загружает в него web-страницу. В переменной **$!** хранится PID последнего фонового процесса, запущенного из данной оболочки, так что в следующих трёх строчках мы можем использовать полученное значения для подгрузки трёх новых вкладок в том же самом окне *Konqueror*.

Другой способ получить нужные номера – это использование правил подстановки, подобных правилам подстановки в оболочке командной строки, примерно так:

```
dcop konqueror-*
```

Эта команда вернёт список PID всех экземпляров *Konqueror*. Если вы запускаете команду или содержащий ее сценарий при помощи горячей клавиши, вы можете захотеть выбрать активное окно, примерно так:

```
for IFACE in $(dcop konqueror-*)
do
if [ $(dcop ${IFACE} konqueror-mainwindow#1 isActiveWindow) == "true" ]
then
MYIFACE=${IFACE}
break
fi
done
```

Этот сценарий перебирает все экземпляры *Konqueror* до тех пор, пока не найдёт активный. Чтобы узнать, как это делается, вам потребуется KDCOP, поскольку для разных приложений эта команда разная.

GUI к оболочке командной строки

Теперь, когда вы получили общее представление о DCOP, позвольте мне показать несколько интересных примеров его использования. Спросите у здравых пользователей Linux, что они предпочитают – GUI или командную строку, и они ответят, что пользуются и тем и другим, переключаясь по мере надобности. Вот небольшой сценарий DCOP, упрощающий эту задачу. На вашем месте я бы сохранил его, привязав к горячей клавише для запуска при работе с *Konqueror* как файловым менеджером.

```
#!/bin/sh
#Включите приведенный выше код для поиска интерфейса Konqueror
if [ -n «${MYIFACE}» ];
then
konsole --workdir $(dcop ${MYIFACE} konquerormainwindow#1
currentURL | cut -c 8-)
fi
```

Konqueror возвращает текущий путь в виде полного URL, но Konsole не понимает присутствующего в его начале '**file://**', так что мы удаляем его командой cut.

Попробуем теперь рассмотреть второй пример. Я написал эту статью в *Kate*, текстовом редакторе из KDE. В целом *Kate* мне нравится, но серьёзным упущением является отсутствие счётчика слов (не говорите мне, что это редактор программного кода, у него же есть проверка орфографии, так ведь?). К счастью, благодаря DCOP добавить подсчёт слов очень просто.

```
#!/bin/sh
for IFACE in $(dcop kate-*)
do
 if [ $(dcop ${IFACE} __KateMainWindow#1 isActiveWindow) == "true" ]
 then
 MYIFACE=${IFACE}
 break
 fi
done
if [ -z "${MYIFACE}" ]
then
 RESULT="No active Kate window found"
else
 ALL=$(dcop ${MYIFACE} EditInterface#1 text | wc -w)
 SELECT=$(dcop ${MYIFACE} SelectionInterface#1 selection | wc -w)
 RESULT="All text: ${ALL} words\nSelected: ${SELECT} words"
fi
kdialog --title "Kate word count" --msgbox "$RESULT"
```

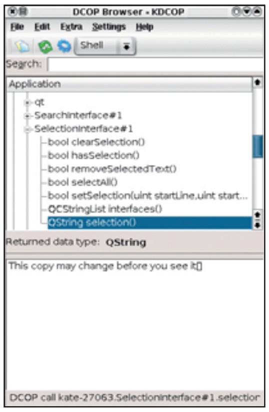
Этот сценарий находит активное окно *Kate* (если такое есть), а затем, с помощью функций **EditInterface** и **SelectionInterface** читает весь текст в окне и в выбранном отрывке. Утилита *dcop* отправляет полученные данные в стандартный поток вывода, так что сценарий может перенаправить его в утилиту *wc* для подсчёта числа слов. Как всегда, проще всего вызывать такой сценарий с помощью горячей клавиши.

Другие языки

Мы с вами рассмотрели, как использовать *dcop* для управления приложениями KDE из командной строки или сценария оболочки, который можно вызывать по горячей клавише. Но поскольку протокол DCOP был придуман для того, чтобы облегчить взаимодействие программ между собой, его можно использовать во многих других языках программирования. Графическое приложение KDCOP может показать синтаксис вызова для Python и C++ точно так же, как для сценария оболочки. Если вы хотите использовать DCOP в своей программе для вызова других приложений или же хотите предоставить доступ извне к функциям вашего, изучите официальную документацию (с множеством примеров) на **http://developer.kde.org/documentation/other/dcop.html**.

Если вы – счастливый обладатель пульта дистанционного управления (например, от TV-тюнера) и *lirc*, перед вами открывается море новых возможностей. KDE позволяет назначить ответные действия для небольшого числа приложений, в основном, мультимедиа-проигрывателей – ищите их в KDE Control Center. Там же можно включить KDE Program Launcher, посредством которого легко запустить утилиту *dcop* и управлять приложениями KDE с пульта ДУ!

KDE иногда критикуют за то, что в нем слишком много опций. Но сколько бы возможностей ни предусмотрел автор программы, всегда может оказаться, что чего-то не хватает. К счастью, благодаря DCOP в KDE вы можете самостоятельно добавить то, что вам нужно. LXF



KDOP – это простой и удобный способ посмотреть список команд DCOP приложения и узнать, что они делают.

ПОДСКАЗКА

- Сначала проверьте работу команды из KDCOP, потом из командой строки и только потом назначайте ей горячую клавишу.
- Если ваш сценарий или команда требуют наличия активного окна, при тестировании их из командой строки используйте следующий способ запуска:

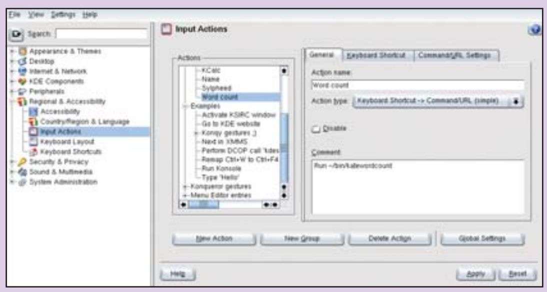
```
sleep 3; ./scriptname
```

- В этом случае у вас будет три секунды на то, чтобы переключиться в нужное вам окно.
- Некоторые интерфейсы содержат очень много функций, например в главном окне *Konqueror* их более 150. Если запустить *dcop* с неполным набором аргументов, он покажет полный список доступных вариантов. Его можно использовать для поиска нужной функции. Например, следующая команда покажет все функции, в названии которых содержится слово "url":

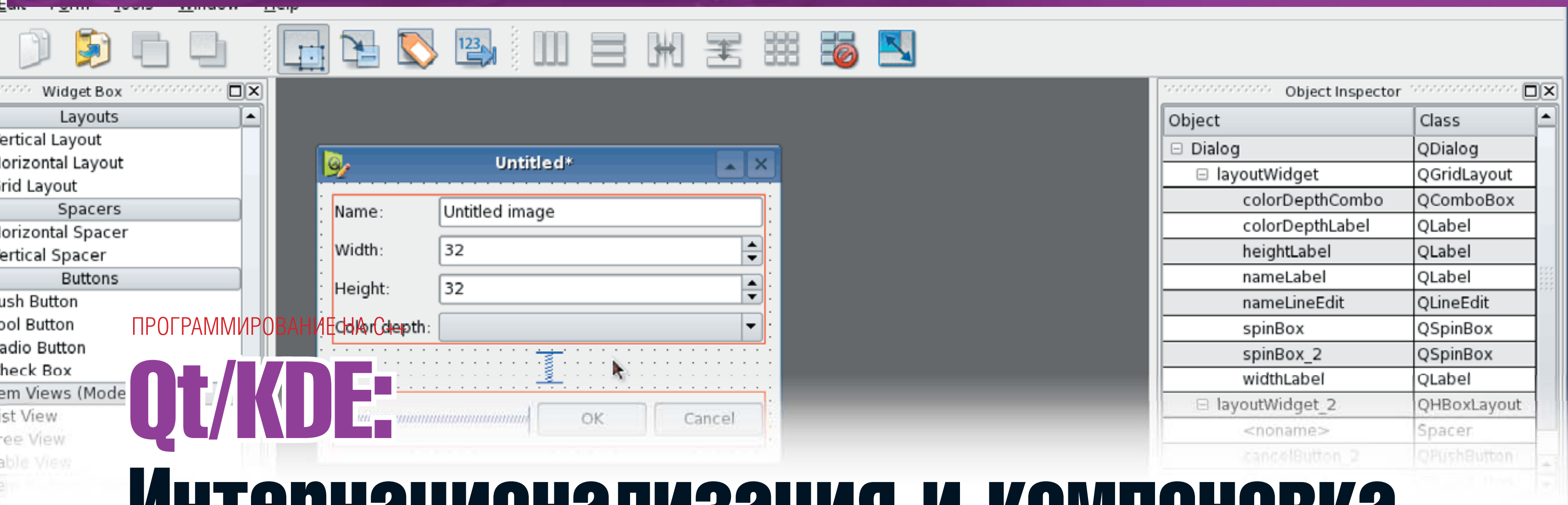
```
dcop konqueror-$! konqueror-mainwindow#1 | grep -i url
```

ПРИВЯЗКА КОМАНД К ГОРЯЧИМ КЛАВИШАМ

Программа *KHotKeys* из состава KDE позволяет привязать выполнение команды к горячей клавише или мышиному жесту. Привязать можно любую команду, будь то вызов dcop или сценарий оболочки с целым списком команд для выполнения. Запустите KDE Control Centre и перейдите на страницу Regional & Accessibility > Input Actions, выберите группу Commands и нажмите кнопку New Action. Осталось выбрать горячую клавишу, указать путь к вашему сценарию – и готово! В системе есть много различных примеров, установленных, но отключенных, так что вы можете посмотреть, что ещё можно сделать.



Раздел Input Actions в KDE Control Centre позволит вам выполнить любую команду по нажатию горячей клавиши.



ПРОГРАММИРОВАНИЕ НА С++

Qt/KDE: Интернационализация и компоновка

ЧАСТЬ 2 Parlez-vous Français? Ваше приложение в любой момент могут запустить в интернет-кафе где нибудь на Монмартре, поэтому говорить и выглядеть нужно соответствующе. Оказывается, это вовсе не так уж трудно – следуйте за **Андреем Боровским**.

МЕСЯЦ НАЗАД

Мы рассмотрели азбуку Qt – структуру приложения, базовые элементы интерфейса, а также очень мощную концепцию слотов и сигналов.

В мире, где приложения распространяются через Интернет и становятся моментально доступны пользователям по всему земному шару, качественно выполненная интернационализация может существенно способствовать успеху продукта. Пример буквально у нас перед глазами. Как вы думаете, велико ли было бы число пользователей Linux, если бы эта система не обрела многоязычный интерфейс? Но интернационализация приложения – задача не простая и проблема тут не только в том, что надо знать иностранные языки. Когда величайшие мудрецы человечества учили свои народы грамоте, они, к сожалению, не предвидели, что в далеком будущем их потомки будут вводить текст с клавиатуры. Мало того, что в одних языках пишут слева направо, а в других – справа налево, начертание символов в некоторых языках (например, арабском) зависит от соседних символов, так что при подсчете ширины текста в пикселях приходится учитывать правописание. Мы упростим себе задачу и рассмотрим интернационализацию на примере европейских языков. Напишем небольшую программу, которая в дальнейшем подвергнется интернационализации:

```
#include <qapplication.h>
#include <qpushbutton.h>
int main(int argc, char *argv[])
{
 QApplication i18nApp(argc, argv);
 QPushButton button(QObject::tr("Exit"), NULL);
 QObject::connect(&button, SIGNAL(clicked()), &i18nApp, SLOT(quit()));
 i18nApp.setMainWidget(&button);
 button.show();
 return i18nApp.exec();
}
```

Главным и единственным визуальным элементом является кнопка button. Ее сигнал **clicked** связан со слотом **quit** объекта **i18nApp**, так что щелчок по кнопке приводит к завершению работы приложения. Единственным текстом, подлежащим интернационализации, является надпись на кнопке – «Exit». По сравнению с предыдущим уроком, в программу добавился лишь один новый элемент – вызов статического метода **tr()** класса **QObject**. Метод **tr()** выполняет замену переданной ему строки на перевод, соответствующий локализации системы, если, конечно, эта строка содержится в ресурсах программы. Таким образом, нам нужно решить две задачи: создать файлы ресурсов (по одному файлу для каждого поддерживаемого языка), необходимые для работы метода tr() и добавить в программу код, отвечающий за загрузку этих ресурсов во время ее выполнения. На практике этот процесс состоит из нескольких шагов. Сохраните текст программы в файле **app2.1.cpp**, и вызовите последовательность команд **qmake**, необходимую для генерации make-файла. Помимо самого make-файла, будет сгенерирован файл проекта с расширением **.pro**. В него-то и нужно добавить ссылки на файлы, которые будут содержать ресурсы интернационализации. Откроем файл проекта в текстовом редакторе и добавим в него следующие строки:

```
TRANSLATIONS=app21_ru.ts\
 app21_fr.ts
```

Мы создали в файле проекта раздел TRANSLATIONS (переводы) и добавили в этот раздел ссылки на два (пока еще не существующих) исходных файла переводов – **app21_ru.ts** и **app21_fr.ts**, для русского и французского языков, соответственно. Имена этих файлов могут быть любыми, но очень удобно включать в них в качестве суффикса или префикса название соответствующей локали (**ru** – для русской и **fr** – для французской) – дальше мы увидим, почему. Если вы захотите перевести интерфейс программы и на другие языки, вам понадобятся дополнительные файлы с именами, соответствующими той же схеме. После того, как мы добавили ссылки на файлы перевода в файл проекта, мы можем сгенерировать сами файлы с помощью утилиты **lupdate**.

В окне консоли дайте команду:

```
lupdate app2.1.pro
```

где **app2.1.pro** – файл проекта. В результате будут созданы файлы **app21_ru.ts** и **app21_fr.ts**. Теперь, когда у нас появились заготовки файлов переводов, мы можем приступить непосредственно к процессу. Для этой цели нам понадобится графическая утилита *Qt Linguist* (рис. 1).

Интерфейс этой утилиты самоочевиден. Запустив *Qt Linguist*, мы последовательно открываем созданные нами файлы с расширением **.ts** и выбираем в левом окне объект создаваемой программы, предна-

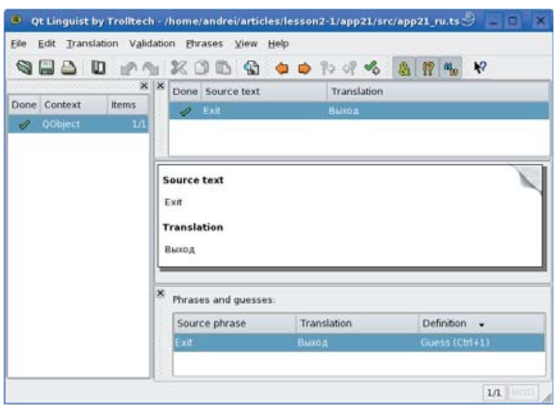


Рис. 1. *Qt Linguist.*

значенный для перевода. При этом в главном окне утилиты отображаются все связанные с объектом строковые ресурсы. В нашем случае мы имеем дело только с одним объектом-потомком **QObject**, у которого есть только один строковый ресурс – «Exit». В файле **app21_ru.ts** мы сопоставляем строке «Exit» перевод «Выход», после чего сохраняем файл. Аналогичные операции мы выполняем с файлом **app21_fr.ts**, в котором переводим строку «Exit» как «Quiter». Итак, у нас есть два файла, содержащие перевод, но наша работа на этом далеко не закончена. Созданные нами файлы с расширением ***.ts** можно рассматривать как исходные тексты файлов ресурсов перевода. Эти ресурсы еще нужно скомпилировать с помощью утилиты **lrelease** (те, кто выполнял интернационализацию приложений в версиях **Qt 2.x**, могут заметить, что порядок вызова утилит изменился). В окне терминала дайте команду:

```
lrelease app2.1.pro
```

В результате у нас появятся два новых файла: **app21_ru.qm** и **app21_fr.qm**. Это и есть двоичные файлы ресурсов интернационализации. Для завершения интернационализации нашей маленькой программы нам осталось только добавить в нее код, выполняющий загрузку этих ресурсов во время выполнения. Рассмотрим теперь текст нашей программы целиком:

```
#include <qapplication.h>
#include <qpushbutton.h>
#include <qtranslator.h>
#include <qtextcodec.h>
int main(int argc, char *argv[])
{
 QApplication i18nApp(argc, argv);
 QTranslator translator(0);
 QString fileName = QString("app21_") + QTextCodec::locale();
 translator.load(fileName, ".");
 i18nApp.installTranslator(&translator);
 QPushButton button(QObject::tr("Exit"), NULL);
 QObject::connect(&button, SIGNAL(clicked()), & i18nApp, SLOT(quit()));
 i18nApp.setMainWidget(&button);
 button.show();
 return i18nApp.exec();
}
```

Класс *QTranslator*, объявленный в файле **qtranslator.h**, загружает файлы ресурсов интернационализации во время выполнения программы и управляет переводом интерфейса на соответствующий язык. Как определить, какой файл ресурсов должен быть загружен? Класс *QTextCodec* обладает статическим методом **locale()**, который возвращает строку, содержащую имя текущей локали. Поскольку мы использовали имя локали в качестве суффикса в именах файлов ресурсов, для получения имени нужного файла мы просто добавляем имя локали к шаблону **«app21_»** и сохраняем результат в переменной **fileName**. Выбранный файл ресурса с расширением qm загружается с помощью метода **load** объекта **translator**. Метод **installTranslator()** класса *QApplication* устанавливает объект, выполняющий перевод интерфейса приложения. Теперь метод **QObject::tr()** может выполнять свою работу. Кстати, возьмите себе за правило использовать **tr()** для всех строк, которые может увидеть пользователь, даже если на текущем этапе вы не планируете интернационализацию приложения – это здорово упростит вашу задачу в дальнейшем. Конечно, если вы работаете со строкой внутри объекта-потомка **QObject** (а так, скорее всего, и будет), **QObject::tr()** можно сократить до **tr()**.

На этом интернационализация программы закончена. Если вы все сделали правильно, то после запуска в окне терминала скомпилированной программы в русскоязычной среде *Qt* вы увидите текст кнопки на русском языке (рис. 2). Программа будет «говорить» по-русски в русскоязычной локализации *Qt*, по-французски – во французской, а во всех остальных локализациях будет использоваться английский текст, заданный по умолчанию. Обратите внимание, что файлы ресурсов интернационализации не становятся частью исполняемого файла приложения – их следует распространять вместе с ним. Это разумно, поскольку для проекта, переведенного на многие языки, таких файлов может быть много,

ЛИНГВИСТУ НА ЗАМЕТКУ

Как уже упоминалось, утилиты семейства *Qt Linguist* используют вызовы **tr()** как маркеры для строк, подлежащих переводу. Как быть, если строка сама по себе нестатическая? Например:

```
QString message = "Host " + host + "not found";
label.setText(message)
```

Естественно, конструкция tr(message) не сработает – мы совсем не заинтересованы в том, чтобы переводить имя переменной. **tr("Host ") + host + tr("not found")** тоже не лучшее решение – разрывается целостность фразы и переводчик может не понять, о чем идет речь. Правильный вариант – использовать подстановки и метод **arg()** класса **QString**:

```
QString message = tr("Host %1 not found");
label.setText(message.arg(host))
```

а в каждой конкретной системе понадобится только несколько (как правило, – всего один) из них. Файлы интернационализации должны располагаться в рабочей директории программы, вот почему при отладке программу следует запускать из терминала, а не из графической оболочки (для которой лучше всего создать специальный ярлык).

Одна из проблем, которую мы не затронули в нашем простом примере заключается в том, что утилита *lupdate* обрабатывает только тот текст, который помечен для перевода (в качестве «маркера» может выступать вызов метода **tr**). Но как отметить статический текст, заданный вне тела функции, например, строковые константы? Для этого в Qt реализованы специальные макросы **QT_TR_NOOP** и **QT_TRANSLATE_NOOP**. Эти макросы служат маркерами для *lupdate*, причем более длинный вариант макроса позволяет определить так называемый контекст для константы, тогда как более короткий предполагает текущий контекст.



Рис. 2. Русскоязычный интерфейс.

Управление внешним видом приложения

Интерфейс любого «настоящего» приложения содержит множество визуальных элементов, расположенных в главном и вспомогательных окнах наиболее удобным для пользователя образом. До сих пор мы сталкивались только с одним элементом управления расположением визуальных элементов – классом *QVBox*, который располагает дочерние элементы вертикально друг под другом. Библиотека *Qt* предоставляет также классы *QHBox* и *QGrid*, позволяющие расположить дочерние элементы соответственно горизонтально и в виде таблицы. Хотя, возможно, это и не очевидно, указанные классы могут служить основой самого сложного интерфейса. Наглядный пример, – редактор *OpenOffice.org*, в котором я пишу этот текст. В первом приближении интерфейс редактора содержит пять элементов, расположенных вертикально: строку главного меню, две панели инструментов (их может быть и больше, и меньше), окно ввода текста и строку состояния в нижней части окна. Фактически, основой интерфейса для такой сложной программы мог бы стать элемент *QVBox*! Конечно, в реальности все не так просто, и *QVBox* не обладает всей необходимой функциональностью. Однако этот пример демонстрирует основную идею – построение сложных интерфейсов начинается с простых элементов управления. Мы напишем приложение просмотра шрифтов, обладающее довольно сложным интерфейсом и для этого тоже начнем с простого элемента – главного окна, которое станет контейнером для всех графических элементов приложения. Такое окно предоставляет класс *QDialog*. Вопреки своему названию, этот класс создает простое окно общего назначения, а не так называемое «диалоговое окно».

Разработка приложений *Qt* немыслима без объектно-ориентированного программирования, к которому теперь прибегнем и мы. Весь интерфейс и логика работы нашего приложения будут содержаться в классе *MainForm*, потомке класса *QDialog*. Чтобы стало понятно, что мы делаем, лучше всего сразу взглянуть на главное окно нашей программы (*рис. 3*). В окне также отмечены границы элементов интерфейса, о которых речь пойдет ниже.

>>

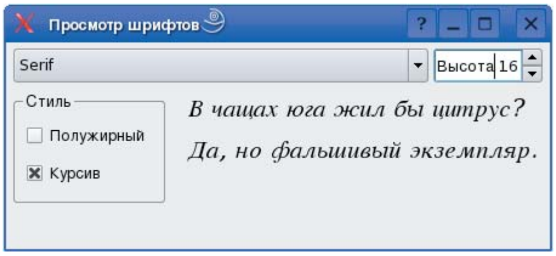


Рис. 3. Программа для просмотра шрифтов.

Декларацию главного класса *MainForm* сохраним в файле **app2.2.h**:

```
#ifndef APP2_2_H
#define APP2_2_H
#include <qdialog.h>
#include <qlayout.h>
#include <qspinbox.h>
#include <qcombobox.h>
#include <qvgroupbox.h>
#include <qcheckbox.h>
#include <qlabel.h>
class MainForm : public QDialog
{
 Q_OBJECT
public:
 MainForm( QWidget* parent = 0, const char* name = “”,
 bool modal = FALSE, WFlags f = 0 );
 ~MainForm() {}
private:
 QVBoxLayout * MainFormLayout;
 QHBoxLayout * ChooseFontLayout;
 QHBoxLayout * ViewTextLayout;
 QComboBox * SelectFontBox;
 QSpinBox * SetSize;
 QVGroupBox * SetFontGroup;
 QCheckBox * SetBold;
 QCheckBox * SetItalic;
 QLabel * TextLabel;
private slots:
 void setLabelFont();
};
#endif // APP2_2_H
```

а реализацию методов – в файле **app2.2.cpp**:

```
include <qfontdatabase.h>
#include <qfont.h>
#include “app2.2.h”
 MainForm::MainForm(QWidget* parent, const char* name,
 bool modal, WFlags f) : QDialog(parent, name, modal, f)
{
 setCaption(trUtf8(“Просмотр шрифтов”));
 resize(300, 200);
 MainFormLayout = new QVBoxLayout(this, 5, 5);
 ChooseFontLayout = new QHBoxLayout(MainFormLayout, 4);
 SelectFontBox = new QComboBox(FALSE, this);
 ChooseFontLayout->addWidget(SelectFontBox, 0, Qt::AlignTop);
 SetSize = new QSpinBox(8, 72, 1, this);
 SetSize->setPrefix(trUtf8(“Высота “));
 SetSize->setValue(12);
 ChooseFontLayout->addWidget(SetSize, 0, Qt::AlignTop);
 QSizePolicy fixedSP = QSizePolicy(QSizePolicy::Fixed, QSizePolicy::
Fixed, FALSE);
 SetSize->setSizePolicy(fixedSP);
 ViewTextLayout = new QHBoxLayout(MainFormLayout, 4);
 SetFontGroup = new QVGroupBox(this);
 SetFontGroup->setTitle(trUtf8(“Стиль”));
 SetFontGroup->setSizePolicy(fixedSP);
 SetBold = new QCheckBox(SetFontGroup);
 SetBold->setText(trUtf8(“Полужирный”));
 SetItalic = new QCheckBox(SetFontGroup);
 SetItalic->setText(trUtf8(“Курсив”));
 ViewTextLayout->addWidget(SetFontGroup, 0, Qt::AlignLeft | Qt::
AlignTop);
 TextLabel = new QLabel(this);
 TextLabel->setAlignment(Qt::AlignLeft);
 TextLabel->setText(trUtf8(“В чащах юга жил бы цитрус?\nДа, но
фальшивый экземпляр.”));
 ViewTextLayout->addWidget(TextLabel, 0, Qt::AlignTop);
 MainFormLayout->addStretch(1);
 connect(SelectFontBox, SIGNAL(activated(int)), this,
SLOT(setLabelFont()));
 connect(SetSize, SIGNAL(valueChanged(int)), this,
SLOT(setLabelFont()));
 connect(SetBold, SIGNAL(toggled(bool)), this, SLOT(setLabelFont()));
 connect(SetItalic, SIGNAL(toggled(bool)), this, SLOT(setLabelFont()));
 QFontDatabase fontDatabase;
 SelectFontBox->insertStringList(fontDatabase.families());
 setLabelFont();
}
void MainForm::setLabelFont()
{
 QString family = SelectFontBox->currentText();
 int size = SetSize->value();
 int weight = QFont::Normal;
 if(SetBold->isChecked()) weight = QFont::DemiBold;
 bool italic = SetItalic->isChecked();
 QFont font(family, size, weight, italic);
 TextLabel->setFont(font);
}
```

Рассмотрим конструктор класса *MainForm*. Он передает свои параметры конструктору базового класса. Уже в первой строке конструктора мы видим новый метод: **trUtf8()**. Этот метод представляет собой вариант уже знакомого нам статического метода **tr()**. Метод **trUtf8()** заменяет строку в кодировке Utf8 соответствующей ей строкой перевода в кодировке Utf16 и обычно используется для интернационализации приложений, но в данном случае мы применим его с другой целью. Хотя создаваемое нами приложение, безусловно, может претендовать на всемирную популярность, заниматься его интернационализацией мы не будем: вместо этого мы сразу вводим весь поясняющий текст интерфейса на русском языке. Однако при этом возникает одна проблема, характерная для дистрибутивов SUSE, *Fedora* и их производных. По умолчанию в этих дистрибутивах используется кодировка Utf8, и для того, чтобы русский текст отображался правильно, необходимо явным образом перекодировать его в кодировку Utf16, которую использует Qt. В решении этой проблемы нам на помощь приходит метод **trUtf8()**. Как уже упоминалось, этот метод выполняет перевод интерфейса. Но если ресурс для перевода отсутствует, **trUtf8()** просто перекодирует строку из Utf8 в Utf16, используя статический метод **fromUtf8**. Эту особенность метода trUtf8 мы и используем для перекодировки всего статического текста в нашей программе. Естественно, если бы весь текст набирался на латинице, проблема с кодировками не возникла бы. В дистрибутивах, использующих другие кодировки, явное перекодирование не потребуется.

Далее мы создаем объект *MainFormLayout* класса *QVBoxLayout*. Класс *QVBoxLayout* сам по себе не представляет визуального элемен-

НЕСКОЛЬКО СОВЕТОВ ПО ПРОЕКТИРОВАНИЮ ИНТЕРФЕЙСА

Проектируя графические интерфейсы, начинающие разработчики часто совершают одни и те же ошибки, приводящие к непредсказуемым изменениям внешнего вида окон при изменении их размеров. Причина появления этих ошибок одна – разработчик забывает, что с его программой будут работать на мониторах с разным разрешением, и что пользователь может изменить размер окна. Примером этой категории ошибок может служить ситуация, когда при изменении размеров окна все элементы управления сохраняют прежний размер, или наоборот, масштабируются вслед за окном. В первом случае элементы интерфейса оказываются сгрудившимися в углу окна (а то и по середине), во втором – некоторые из них начинают выглядеть уродливо. Какие элементы должны изменяться, а какие следует оставить неизменными? Высоту элементов (за исключением многострочных и других сложных компонентов ввода или просмотра) следует зафиксировать. Постоянную ширину должны иметь кнопки различных типов и наборные счетчики, а у строк и многострочных элементов ввода, а также у раскрывающихся списков, которые могут содержать длинный текст, она должна быть переменной. Разумеется, из этих правил могут быть исключения, но исключения должны иметь серьезные обоснования, а небрежность разработчика к таковым не относится.

Одним из преимуществ Linux являются невысокие требования к ресурсам. Помните, что вашей программой может пользоваться обладатель монитора с небольшой диагональю и, соответственно, невысоким разрешением. Что может быть хуже ситуации, когда размеры окна по умолчанию превышают размеры рабочей области и какая-нибудь важная кнопка оказывается за ее пределами? Если элементов управления много, размещайте их в нескольких окнах, а не в одном.

та. Этот класс относится к менеджерам компоновки (layout managers) и его задача заключается в управлении расположением дочерних визуальных компонентов на родительской форме. Название класса *QVBoxLayout* похоже на название класса *QVBox*, и это не случайно. Класс *QVBoxLayout* управляет дочерними элементами также как *QVBox*, то есть располагает их вертикально, один под другим. Класс *QHBoxLayout* выстраивает элементы горизонтально. Почему мы используем классы *QDialog* и *QVBoxLayout*, а не класс QVBox? Преимущество использования *QVBoxLayout* (как и других классов *–Layout*) заключается в том, что мы можем использовать одновременно несколько таких классов для одного окна. Далее мы увидим, как это делается. Конструктору объекта *MainFormLayout* передается ссылка на окно *MainForm* (this). Таким образом, мы сообщаем классу, что он будет управлять расположением элементов данного окна.

Взгляните на *рис. 4* справа. На нем можно выделить две вертикально расположенные группы, в каждой из которых два элемента интерфейса (на рисунке эти группы обведены синими контурами). В верхней группе расположены раскрывающийся список и наборный счетчик (spinbox), в нижней – флажки и текстовая метка. Чтобы получить эти группы, мы создаем два объекта класса *QHBoxLayout* для горизонтального расположения элементов и делаем их дочерними элементами *MainFormLayout*, чтобы сами группы располагались вертикально (область, которой управляет *MainFormLayout*, обведена красным). Сразу после создания «вертикального менеджера» компоновки *MainFormLayout* мы создаем дочерний по отношению к нему «горизонтальный» менеджер *ChooseFontLayout*. В конструкторе классов *QHBoxLayout* и *QVBoxLayout* мы, помимо прочего, указываем расстояние в пикселях между элементами, находящимися под управлением данных менеджеров. Далее мы создаем объекты классов *QComboBox* (раскрывающийся список), *QSpinBox* и с помощью метода **addWidget()** объекта *ChooseFontLayout* передаем их «под управление» данного объекта. Обратите внимание на флаг **Qt::AlignTop**, который мы передаем методу *addWidget*. Этот флаг указывает на привязку визуального элемента к верхнему краю родительского окна. В результате при изменении размеров окна положение элемента относительно верхнего края не изменится. Другие флаги из этой группы включают **Qt::AlignLeft**, **Qt::AlignRight**, **Qt::AlignBottom** и **Qt::AlignCenter**.

Далее мы выполняем еще одну операцию, связанную с расположением элементов. Наши раскрывающийся список и счетчик будут выстроены горизонтально объектом класса *ChooseFontLayout*. Этот объект позаботится о том, чтобы у элементов, расположением которых он управляет, была одинаковая ширина, и чтобы при изменении размеров окна эта ширина увеличивалась пропорционально. Но нас такое поведение не устраивает. Ширина раскрывающегося списка может меняться вместе с шириной окна, но ширину счетчика желательно оставить такой, какой она была сразу после его создания. Как изменить поведение элементов интерфейса? В процессе создания визуальных объектов Qt автоматически выбирает для них оптимальную ширину, соответствующую так называемой «политике изменения размера» (size policy), заданной по умолчанию. Политика изменения размера представляет собой набор правил, определяющих изменение высоты и ширины элемента интерфейса при изменении размера его родительского окна. «Политика изменения размера» раскрывающегося списка, заданная по умолчанию, нас устраивает, а вот для счетчика лучше установить другую «политику», задающую минимально допустимую (с учетом ширины поля ввода и длины поясняющего текста) ширину, которая не должна меняться при изменении размеров окна. Для этого мы создаем объект **fixedSP** специального класса *QSizePolicy*, инкапсулирующего «политику изменения размера». В конструкторе объекта мы указываем, что ширина и высота, определяемые создаваемой «политикой», должны быть фиксированы. Далее, с помощью метода **setSizePolicy()**, мы указываем объекту *SetSize* новую «политику». Теперь при изменении размеров окна ширина счетчика изменяться не будет, а раскрывающийся список *SelectFontBox* займет все пространство слева от него.

Далее мы создаем второй объект класса *QHBoxLayout* и «упаковываем» в него группу флажков и текстовую метку. Группа флажков представлена объектом *SetFontGroup* класса *QVGroupBox*. Этот класс представляет собой стандартную группу кнопок (напомню, что флажки *QCheckBox* представляют собой разновидность кнопок – сам класс *QCheckBox* является потомком класса *QButton*), снабженную заголовком и рамкой. Буква V в имени класса указывает, что дочерние кнопки будут располагаться вертикально. Для объекта *SetFontGroup* мы указываем ту же «фиксированную» политику изменения размера, что и для счетчика. Добавить флажки в группу очень просто – для этого в конструкторе каждого объекта класса *QCheckBox* нужно указывать в качестве родительского элемента *SetFontGroup*. Далее мы добавляем *SetFontGroup* в число элементов, управляемых *ViewTextLayout*. Из визуальных элементов нам осталось добавить только метку, что мы и делаем.

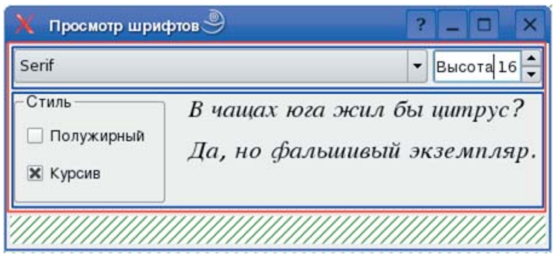


Рис. 4. Схема компоновки главного окна нашего приложения.

Далее следует еще один трюк работы с интерфейсом. С помощью метода **addStretch()** мы добавляем пустое пространство, указав ему фактор растяжения (stretch factor), равный единице. Поскольку у двух объектов *QHBoxLayout* этот фактор равен **0** (значение по умолчанию), при изменении высоты окна вся нижняя часть будет занята пустым пространством (на *рис. 3* эта область заполнена зеленой штриховкой), а элементы управления окажутся «прижатыми» кверху (поэтому stretch иногда называют «пружиной» или «распоркой»). Чтобы лучше понять принцип действия этого механизма, закомментируйте строку **addStretch()**, откомпилируйте приложение и затем попробуйте изменить размер окна.

Теперь на время покинем конструктор и посмотрим на декларацию класса *MainForm*. В классе мы объявляем новый слот. В этом нет ничего сложного, ведь слоты, – это обыкновенные методы классов. Нужно только указать ключевое слово slots после спецификатора видимости и не забыть макрос Q_OBJECT в самом начале объявления класса [Конечно, сам класс при этом должен быть прямым или непрямым наследником **QObject**, – прим.ред.]. Наш слот **setLabelFont()** будет реагировать на сигналы, поступающие от всех элементов управления программы. Когда состояние одного из элементов меняется, метод *setLabelFont* считывает значения всех элементов управления, конструирует объект класса QFont (шрифт) и передает новый шрифт метке. В конструкторе *MainForm* мы связываем слот с четырьмя сигналами, информирующими нас об изменении состояния соответствующих элементов управления. Далее мы создаем список установленных в системе шрифтов с помощью объекта класса *QFontDataBase* и передаем список названий шрифтов объекту *SelectFontBox* с помощью метода **insertStringList()**. Наконец, мы явным образом вызываем **setLabelFont()** для того, чтобы установить для метки шрифт, соответствующий высоте и стилю, заданным по умолчанию (имя этого шрифта будет соответствовать первому шрифту в списке *QFontDataBase*). Ну вот и все... Нет, не все! Мы забыли самую малость, – функцию main для нашей программы. Создадим файл **main.cpp** следующего содержания:

```
#include <qapplication.h>
#include "app2.2.h"
int main(int argc, char *argv[])
{
 QApplication app(argc, argv);
 MainForm mf(0, "MF");
 app.setMainWidget(&mf);
 mf.show();
 return app.exec();
}
```

вот теперь – действительно всё.

Наш далеко не самый сложный пример демонстрирует, кроме прочего, насколько трудоемким может быть программирование интерфейса. В следующей статье мы познакомимся с *Qt Designer* – инструментом, упрощающим процесс построения GUI, а также с несколькими полезными классами библиотеки *Qt*. LXF

ЧЕРЕЗ МЕСЯЦ

Мы научимся создавать интерактивные графические интерфейсы одной левой... кнопкой мыши.



HARDCORE LINUX СЕРИЯ ДЛЯ СПЕЦИАЛИСТОВ

Командная строка: секреты Bash

«Это кронтаб. Детки роются в кронтабе». Примерно так бормочет **Пол Хадсон (Paul Hudson)**, весь этот месяц доказывавший своё мужество в неравной схватке с управлением заданиями, навигацией по каталогам Linux и переменными окружения.

МЕСЯЦ НАЗАД



Мы охватили псевдонимы и условные операторы в Bash и по-новому увидели старых любимчиков less, find и grep.

Тропа гуру командной строки со всех сторон стеснена недочетами графических инструментов и тиранией невежества. Но если сложности прошлого урока не сбили вас с пути, то осталось изучить последние несколько строк катехизиса команд, и вы получите право хвастаться перед окружающими – и согласитесь, это превесело: быть местным парнем-всезнайкой.

В последнем выпуске мы рассмотрели специфические команды Unix-окружения, вместе с некоторыми, встроенными в *Bash*. На этот раз я сосредоточусь на том, что предоставляет *Bash* и как работает система управления заданиями в Unix. Для этого вам могут понадобиться привилегии суперпользователя.

Основы заданий в Bash

Откройте окно терминала и введите такую команду:

```
yes > /dev/null
```

Команда *yes* выводит на экран (или в **/dev/null**, как в нашем примере) 'y', пока вы не прервёте ее (некоторым потребуется время, чтобы понять, в чем ее полезность). Предоставленная сама себе, команда будет выполняться вечно, так что оставим её транжирить циклы процессора, а сами откроем новый терминал и запустим в нём **top**. Вы должны увидеть, что yes использует до 100% процессорного времени. Затем оставьте окно с *top* запущенным и вернитесь в терминал с *yes*. Вам пора уже знать, что нажатие Ctrl+C посылает текущему процессу сигнал **SIGINT** и обычно приводит к его прерыванию. Можете также нажать Ctrl+\, чтобы послать процессу сигнал **SIGQUIT**, сигнал немедленно завершит процесс и создаст дамп памяти (core dump). Впрочем, куда важнее, что вы можете использовать Ctrl+Z, чтобы послать **SIGSTOP**.

Большинство сигналов в Linux отправляются программе и обрабатываются ею. Например, нажатие Ctrl+C посылает **SIGINT** процессу, а тот его обрабатывает. Процесс может проигнорировать сигнал (в этом случае Ctrl+C ничего не делает), а может выполнить некоторые действия перед завершением работы. Но два сигнала – **SIGSTOP** и **SIGKILL** – процессу не отправляются. Вместо этого они передаются непосредственно ядру, потому что требуют внешнего воздействия на

процесс. **SIGKILL** вы уже должны знать по команде **kill -9**; это сигнал, говорящий: «Любой ценой уничтожь этот процесс». **SIGSTOP** же говорит: «Хочу временно приостановить этот процесс. Не прибить его, но и не давать ему процессорного времени».

Давайте попробуем применить это к команде *yes*. Выбрав данный терминал, нажмите Ctrl+Z. Вы должны увидеть такой вывод:

```
[1]+ Stopped        yes > /dev/null
```

Это означает, что наша команда (**yes > /dev/null**) остановлена, и на неё можно сослаться как на задание №1. Терминал теперь в ваших руках, так что попробуйте запустить **fg** (foreground) для реанимации *yes*. Внимание: отправка **SIGSTOP** и запуск **fg** похожи на нажатие паузы на вашем CD-плейере, поскольку процесс на самом деле не прекращает существование, а просто временно приостанавливает работу.

Итак, *yes* снова работает. Нажмите Ctrl+Z, чтобы опять приостановить процесс, но теперь введите **bg**, чтобы *Bash* возобновил его работу и оставил его в фоновом режиме – как будто вы запустили **yes > /dev/null &**. Теперь *yes* работает в фоновом режиме, и Ctrl+Z не сможет остановить его. Вместо этого мы должны послать сигнал **SIGSTOP** непосредственно процессу, используя команду:

```
kill -SIGSTOP %1
```

%1 означает, что *Bash* должен послать сигнал заданию №1, т.е. нашей команде *yes*. Если вы опасаетесь, что не упомните все цифровые идентификаторы, самое время познакомиться с *jobs*. У этой команды нет интересных параметров, но она возвращает список всех созданных нами заданий:

```
paul@Susannah:~$ jobs
[1]-     Running     yes > /dev/null &
[2]      Stopped     top
[3]+     Stopped     du /
```

В левом столбце перечислены все запущенные задания, и вы сможете перезапустить любое из них, просто набрав **%** с последующим номером, например, **%1** перезапустит команду *yes*. В истинном Unix-стиле добавление амперсанда (**%1 &**) перезапустит задание №1 в фоновом режиме.

Вы, вероятно, недоумеваете, почему задание №1 имеет знак «минус» после номера, задание №2 не имеет никакого знака, а №3 – знак «плюс». Так *Bash* отмечает, что рассматривается в качестве текущего (+) и предыдущего (-) задания, и вы можете использовать это для краткости. Например, если перевели команду *yes* в фоновый режим, вы можете воскресить её, набрав **%+** (или даже **%%**). Для ссылки на предыдущее задание используйте **%-**.

Автоматизация работы

Управление заданиями превосходно работает, когда вы подключены к терминалу, но что если вы хотите, чтобы программы останавливались и запускались, когда вас нет на месте? Есть три способа сделать это в Linux, каждый из которых решает проблему слегка по-своему.

Первый – команда *at*, позволяющая запланировать программу на запуск в заданное время. Изначально она проектировалась для одноразовых заданий, которые вы хотели выполнить в определённое время (но не сейчас). Второй инструмент – *batch*, он подобен *at*, с тем исключением, что время вы не задаёте – задание просто запустится, когда система не загружена. Лучше всего это подходит для ситуаций, когда ресурсы компьютера используются кем-то ещё, а вам нужно на что-то надолго отвлечься. Последний инструмент – *Cron*, этот планирует задания для регулярного выполнения.

At и *batch* – милые, удобные инструменты, в то время как *Cron* выглядит слегка пугающе. Так что давайте начнём с того, что попроще…

At – это где?

Команда *at* обычно работает интерактивно, то есть вы сначала вызываете программу, указав время, когда задание должно выполниться, затем вводите свои команды и нажимаете Ctrl+D, чтобы сохранить задание. Вот пример «диалога» с *at*:

```
paul@Susannah:~$ at midnight
at> du / > /home/paul/diskusage
at <EOT>
job 4 at 2006-03-24 00:00
```

В первой строке я запускаю *at* и указываю полночь (midnight) как время старта для задания. При этом *at* запустится, и появится приглашение **at>**, показывающее, что можно вводить содержимое задания. Задание, которое я хочу выполнить в полночь – выяснить, сколько дискового пространства используется на моём компьютере, так что я запускаю *du* и перенаправляю её вывод в файл в моём домашнем каталоге. **<EOT>** – это я нажал Ctrl+D, что приводит к сохранению задания, выводу его номера и сообщению, когда оно будет выполнено (сегодня в полночь).

Узнать, какие задания уже поставлены в очередь, вы можете, набрав **atq**, которая распечатает что-то наподобие

```
4    2006-03-24 00:00    a    paul
```

Первое, второе и четвёртое поля – это соответственно номер задания, время, когда оно будет выполнено, и кто его создал, но о третьем поле нужно сказать особо: это приоритет задания. В Linux можно выбрать много очередей заданий, и чем дальше буква от начала алфавита, тем ниже приоритет. Очередь «a» имеет наивысший приоритет, и будет исполнена с нормальным для пользователя значением nice (то есть так быстро, как только сможет).

Если вы дождётесь полуночи, ваше задание будет выполнено, как и планировалось. Но если вы передумаете, используйте команду *atrm*, чтобы удалить задание:

```
atrm 4
```

Есть несколько способов указать время в *at*, и midnight – лишь один из них. Из предопределённых есть tomorrow (завтра), noon (полдень) и teatime (4 часа вечера), но вы можете указывать и точное время, например, 16:00 (те же 4 вечера) или комбинировать эти значения (16:00 tomorrow). Простейший способ – указывать относительное время, используя now (сейчас), например, так:

> «А ЕСЛИ ВЫ ХОТИТЕ, ЧТОБЫ ПРОГРАММЫ ЗАПУСКАЛИСЬ И ОСТАНАВЛИВАЛИСЬ БЕЗ ВАС?»

```
at now + 5 minutes
at now + 3 hours
at now + 4 weeks
```

Если время выполнения для вас не имеет значения, забудьте про *at* и используйте *batch*. Различие заключается в том, что *batch* начнёт выполнять ваши задания сразу же, как только загрузка системы опустится ниже 0,8 (т.е. машина будет не слишком занята). Синтаксис намного проще, поскольку не нужно указывать время: просто наберите **batch**, нажмите Enter, добавьте свои команды и нажмите Ctrl+D, чтобы сохранить задание.

Введя **atq**, вы увидите ваше задание в очереди «B», что означает запуск с более низким приоритетом, чем у других заданий и большинства программ в системе. По этой причине ваше задание стартует только тогда, когда система бездействует, но если оно запустится, а в следующую секунду машину потребует другая работа, ваше задание тихонько переберется в фоновый режим и отдаст ресурсы процессора. Оно не останавливается, но из-за более низкого приоритета получит намного меньше процессорного времени.

Никто не любит вводить одну и ту же команду снова и снова, так что если вы хотите, чтобы *at* или *batch* читали ваши задания из файла, просто используйте **-f имяфайла** перед указанием времени, например, так:

```
at -f myjob.job tomorrow
batch -f myjob.job
```

Нет ничего прекраснее, чем заставить ваш компьютер делать кучу работы за вашей спиной, полностью автоматически, пока вы играете в UT. Волшебной программой, осуществляющей это, является *Cron*, демон

<<

времени. Каждую минуту каждого часа, ежедневно, на протяжении всего года, миллионы демонов Cron по всему миру проверяют, нужно ли что-то сделать, затем, разочаровавшись, снова бездействуют 60 секунд. Не позволяйте вашему демону скучать – пусть он немного порезвится!

Вот работа для Cron

Задания *Cron* можно разделить на два типа: пользовательские и системные. Большинство дистрибутивов Linux поставляются с *Cron*, способным на обе роли, то есть каждый может назначать свои персональные задания, и сама система тоже будет регулярно что-нибудь запускать. Вам потребуются права суперпользователя, чтобы изменять системные задания, поскольку они отвечают за важные вещи – обновление базы данных locale, генерирование базы данных man, скачивание и применение обновлений безопасности, и т.д. Многие из них, вероятно, уже есть в вашем дистрибутиве, и мы довольно скоро их рассмотрим.

Но сначала рассмотрим пользовательский *Cron* – регулярные задания, назначаемые пользователем для выполнения повседневных задач. Например, часто требуются ночные сборки программного проекта. Если вы тестируете KDE и приближается дата выпуска, вы, вероятно, захотите скачивать новый релиз каждый день, компилировать его и тестировать, выискивая ошибки. Загрузка выполняется довольно просто: возьмите *Konstruct*, отредактируйте конфигурационный файл так, чтобы выполнялась проверка последней версии исходного кода в *Subversion*, и пересоберите. Но сборка KDE может потребовать более четырёх часов, в зависимости от числа изменений и скорости вашей машины. Вот и работа для *Cron*!

> «НЕ ПОЗВОЛЯЙТЕ ВАШЕМУ ДЕМОНУ СКУЧАТЬ – ПУСТЬ ОН НЕМНОГО ПОРЕЗВИТСЯ!»

Введите **crontab -e**, чтобы отредактировать ваш персональный crontab (так называется файл *Cron* со списком ваших заданий). Если у вас его ещё нет, он будет создан. Теперь – самая сложная часть: хитрый формат файла crontab. Наберите в нём такую строку:

```
* * * * * du -h / > /home/paul
```

Вместо **paul** укажите ваше собственное имя [пользователя]. Часть **du** и последующие символы – это команда, которую *Cron* должен выполнить, а все эти звёздочки определяют время, когда она должна выполняться. Это смешнее всего! По порядку:

Минута (0-59)
Час (0-23)
День (1-31)
Месяц (1-12)
День недели (0-7, как 0, так и 7 соответствуют воскресенью)

Звёздочка означает «все значения», так что «* * * * *» означает «каждую минуту каждого часа ежедневно каждый месяц, независимо от дня недели». Следующее задание будет выполняться каждую субботу в полдень:

```
0 12 * * 6 du -h / > /home/paul
```

Cron перечитывает свои конфигурационные файлы раз в минуту, но не сразу выполняет все обнаруженные задания. То есть, если сейчас без десяти секунд полночь и вы установите задание на полночь, оно будет выполнено через 24 часа + 10 секунд. [Точнее, при первом после добавления задания запуске выполняется подготовка заданий, а при втором (через минуту) – их выполнение, – прим. перев.]

Помимо указания точного времени для настройки простых заданий, вы можете определить и более сложные выражения, используя дефисы и запятые. Например, можно создать четыре задания *Cron* для запуска сценария в 11, 12, 13 и 14 часов, а можно просто записать задание таким образом:

```
0 11-14 * * * du -h / > /home/paul
```

Запятые позволяют указывать конкретные значения вместо диапазона, так что мы можем сделать то же самое другим способом:

```
0 11,12,13,14 * * * du -h / > /home/paul
```

Но и без этого вы можете создать столько заданий, сколько захотите – просто вводите их, каждое в свою строку, в ваш crontab. Просмотреть текущий crontab позволяет команда **crontab -l** (маленькая L), но она просто распечатает то же самое, что вы увидите при редактировании файла, так что проку от нее мало.

Корни Cron

Раз уж вы освоили мощь *Cron* с точки зрения пользователя, пора браться за более «продвинутую» вещь: редактирование системных файлов crontab. Да, их несколько – во многих системах есть скрипты, содержащие почасовые, ежедневные, еженедельные и даже ежемесячные задания. Откройте в своём любимом текстовом редакторе файл **/etc/crontab**, и вы увидите нечто вроде

```
17 * * * * root run-parts --report /etc/cron.hourly
25 6 * * * test -x /usr/sbin/anacron || run-parts --report /etc/cron.daily
47 6 * * 7 test -x /usr/sbin/anacron || run-parts --report /etc/cron.weekly
52 6 1 * * test -x /usr/sbin/anacron || run-parts --report /etc/cron.monthly
```

То есть часовые задания выполняются в 17 минут каждого часа, ежедневные – каждое утро в 6:25, еженедельные – по воскресеньям в 6:47 и ежемесячные – в 6:52 первого числа месяца. На вид время выполнения задано «с потолка», но в нем есть свой смысл – большинство заданий выполняется утром, когда нагрузка минимальна, и они запускаются в разное время, чтобы избежать столкновения между собой.

В системном файле есть одно дополнительное поле, используемое для указания имени владельца задания. В нашем примере везде стоит root, но вы можете изменить его на более подходящее для вашего собственного перечня.

Если вы не сталкивались с этой утилитой раньше, то поясним, что **run-parts** выполняет все программы в заданном каталоге и может дополнительно предоставлять отчёт (**--report**) об этом выполнении.

КАК AT.ALLOW И AT.DENY РАСПОРЯЖАЮТСЯ ПРАВАМИ НА ЗАДАНИЯ

at.allow	at.deny	Можете ли вы использовать at/batch?
Не существует	Не существует	Нет
Не существует	Существует, и вы в нём	Нет
Не существует	Существует, но вас там нет	Да
Существует, но пуст	Существует, и вы в нём	Нет
Существует, но пуст	Существует, но вас там нет	Нет
Существует, и вы в нём	Существует, но вас там нет	Да
Существует, но вас там нет	Существует, и вы в нём	Нет
Существует, и вы в нём	Существует, и вы в нём	Да

Она идеальна для заданий *Cron*, поскольку может разделять их по каталогам **/etc/Cron.hourly**, **/etc/Cron.daily** и так далее. Если вы хотите добавить собственное задание в список ежечасных, просто скопируйте исполняемый сценарий в **/etc/Cron.hourly** и предоставьте *run-parts* сделать всё остальное.

Помимо настройки системных заданий, права root дают нам ещё одну привилегию: мы можем решать, кому позволить, а кому запретить планирование заданий. Каждый пользователь, имеющий право использовать *at* и *batch*, должен быть указан в файле **/etc/at.allow**. Но если вы попытаетесь открыть этот файл (как root), то, вероятно, обнаружите, что его вообще не существует. Это потому, что по умолчанию каждый пользователь рассматривается как имеющий право работать с *at* и *batch*, если он явно не указан в файле **/etc/at.deny** [который по умолчанию присутствует, – прим. перев.]. В представленной таблице разъясняется, как работают права.

Учитесь запоминать

Вспомните лекции по информатике. В частности, абстрактные типы данных. Вы должны помнить их – стеки, очереди и т.д. Если же вы спали на лекциях (думаю, большинство из нас так и делало), напомню, что стеки – это массивы типа «Последний пришёл – первый вышел», то есть поместив в стек элемент A, затем B, а затем C, извлечь их оттуда вы сможете в порядке C, B, A. Элемент A не достать, пока не извлекут элементы C и B.

Всё это я говорю потому, что мы частенько рассматриваем навигацию по каталогам Linux как совершенно произвольную и сложную для отслеживания операцию, хотя на самом деле можно трактовать наши перемещения как стек. *Bash* выполняет всё это для нас с помощью команд *pushd* и *popd*, которые изменяют текущий каталог. Вы помещаете (push) каталоги в стек *Bash* (перемещаясь в них и добавляя их к списку сохранённых каталогов), и затем извлекаете (pop) их из стека (удаляя его и возвращаясь в предыдущий каталог).

Это лучше продемонстрировать на примере:

```
paul@Susannah:~$ pushd /etc/init.d
/etc/init.d ~
paul@Susannah:/etc/init.d$ pushd /var/log
/var/log /etc/init.d ~
paul@Susannah:/var/log$ pushd /usr/local/bin
/usr/local/bin /var/log /etc/init.d ~
paul@Susannah:/usr/local/bin$ popd
/var/log /etc/init.d ~
paul@Susannah:/var/log$ popd
/etc/init.d ~
paul@Susannah:/etc/init.d$ popd
~
paul@Susannah:~$
```

Первоначально я нахожусь в моём домашнем каталоге, но хочу перейти в **/etc/init.d**. Если я выполню **cd /etc/init.d,** я тут же потеряю путь, где я был до этого, поэтому я использую **pushd /etc/init.d**, которая делает текущим **/etc/init.d**, но сохраняет мой предыдущий каталог в стеке каталогов. Обратите внимание, что *Bash* выводит на экран мой стек – сейчас это **/etc/init.d ~**, что означает: «Ты сейчас в **/etc/init.d**, а до этого был в **~**».

Теперь я снова использую **pushd /var/log**, чтобы сменить каталог, и *Bash* добавит его в список. Далее, я таким же образом перехожу в **/usr/local/bin**, и теперь у нас четыре элемента в стеке каталогов. Пора воспользоваться *popd*. Каждый раз, когда я вызываю *popd*, верхний элемент (каталог слева в стеке) удаляется, и я возвращаюсь в предыдущий каталог. Из **/usr/local/bin** я перехожу в **/var/log**, затем в **/etc/init.d**, и, наконец, назад в **~**, мой домашний каталог – и стек опять пуст.

Есть ещё несколько трюков, которые можно выполнить с помощью *pushd* и *popd*. Во-первых, простой ввод *pushd* будет переключать вас между двумя верхними каталогами в стеке, перемещая вас туда и обратно без изменения остальной части стека. С помощью *popd* вы можете удалить каталоги из стека, введя **popd +X**, где **X** – элемент стека, подлежащий удалению. Нумерация элементов в стеке начинается с нуля, так что самый верхний каталог (текущий) будет иметь индекс 0, следующий – 1, и т.д. Таким образом, ввод **popd +2** удалит из стека третий элемент.

ПОДСКАЗКИ

* Графические терминалы выполняют переключение между терминалами быстрее и не ограничивают вас шестью терминалами.

* Добавьте **-m** к вашей команде *at*, чтобы получить уведомление о выполнении задания по электронной почте.

* *Batch* помещает ваши задания в очередь **B**, но вы можете указать любую очередь, какую пожелаете, используя параметр **-q**, так что **-q d** присвоит вашим заданиям значение **nice**, равное 8 (ещё более низкий приоритет).

* Запрет пользователям запускать *at* (их отсутствие в **at.allow**, либо присутствие в **at.deny**) не позволит им использовать *at*, *atq*, *atrm*, но не удалит уже запланированные задания.

* Запустите **crontab -r**, чтобы удалить все ваши персональные cron-задания.

* Вы можете просмотреть стек каталогов в любое время, введя **dirs**, или очистить его командой **dirs -c**.

Сохранение окружения

Bash обеспечен полным комплектом переменных окружения – введите **set**, и вы поймёте, о чем я говорю. Многие из них устанавливаются автоматически, такие как **USER** (ваше имя пользователя), **LINES** (число строк, отображаемых вашим терминалом), **HOME** (ваш домашний каталог) и т.д. Некоторые переменные окружения уже установлены, но часто меняются, например, **PATH**, сохраняющая все каталоги, где будет выполняться поиск исполняемых файлов. Другие переменные окружения полностью определяются пользователем – вы можете устанавливать собственные значения, чтобы хранить интересующие вас данные.

> «BASH БИТКОМ НАБИТА ПЕРЕМЕННЫМИ ОКРУЖЕНИЯ - ВВЕДИТЕ SET И УВИДИТЕ.»

Изменение таких переменных, как **PATH**, бывает весьма полезным, особенно учитывая, что *Bash* различает операции «установка переменной для одной команды» и «установка переменной для всех команд в текущем сеансе». Попробуйте сами:

```
cp /bin/mount ~/date
PATH=. date
date
```

Здесь **/bin/mount** копируется в ваш домашний каталог и переименовывается в *date*. Но *date* – это исполнимая программа в **/bin**, так что второй командой мы говорим: «Установи наш путь только на текущий каталог (так что **~/date** будет читаться, а **/bin/date** нет) и затем исполни команду *date*». В результате мы запустим нашу копию *mount*. Но если вы выполните следующую команду, *date*, вы увидите нормальный вывод программы *date*, поскольку переменная PATH вернётся к своему обычному значению. Если вы хотите установить переменную на весь сеанс, используйте *export*, например, так:

```
export PATH=/path/to/someplace:$PATH
```

Эта команда добавит **/path/to/someplace** в начало списка каталогов в **PATH**, дописав оставшиеся значения в конец.

Некоторые дистрибутивы (самый злостный преступник, пожалуй, SUSE) доводят использование переменных окружения до крайности и записывают в них сложные функции, загромождающие вывод простой и милой команды set, но не пугайтесь: воспользуйтесь графическим терминалом, установите буфер вывода побольше и прокручивайте окно назад, пока не найдёте то, что вас интересует. LXF

ЧЕРЕЗ МЕСЯЦ

Любите вы их или ненавидите, но регулярные выражения существуют, так почему бы не почитать наш мастер-класс?



ПРОГРАММИРОВАНИЕ

PHP Что новенького в PHP 6

Что PHP грядущий нам готовит? **Пол Хадсон (Paul Hudson)** всегда на острие технологий и собирается показать нам, как использовать новые возможности.



МЕСЯЦ НАЗАД

Мы рассматривали представления, поскольку не все пользователи базы данных обладают такими же привилегиями, как мы – от некоторых вещей их стоит держать подальше для их же собственной пользы.

Не только пионер должен быть всегда готов! В нашем мире программист программисту волк, так что не стоит пренебрегать ни одной возможностью усовершенствовать свой код. PHP 5.1 позволил подкрутить доступ к данным с помощью PDO и преподнёс множество маленьких ускорений по всей системе, но более крупные и всеобъемлющие изменения остались за горизонтом. Часть из них запланирована на версию 5.2, остальные – на PHP 6, но и те, и другие уже не за горами. Думаю, что релиз PHP 5.2 можно полностью пропустить, сосредоточившись на PHP 6, который уже совсем близко.

От изменения первой цифры версии мы обычно ждем нарушений обратной совместимости, множества новых возможностей и некоторых сюрпризов. В этой статье я собираюсь сосредоточиться на том, как PHP 6 может повлиять на ваши сценарии, и какие изменения потребуется в них внести, чтобы решить проблемы ещё до их появления. Но, пожалуйста, не забывайте, что PHP 6 ещё не выпущен, и изменения, о которых мы говорим тут, могут сами претерпеть изменения (и, скорее всего, так и будет) перед финальным релизом.

Универсальный переводчик

PHP – это язык программирования, который управляет сотнями тысяч web-сайтов, но при этом до сих пор, вопреки культурному разнообразию Интернета, очень плохо поддерживает Unicode. Когда вы создаёте новую строку или читаете её из базы данных, скорее всего, вы используете старый добрый ASCII. Существуют расширения для работы с многобайтовыми символами, но они ограничены кое-какими функциями и скромным набором строковых типов, и вы даже не можете гарантировать, что нужное расширение будет установлено на том сервере, где предстоит работать вашему коду.

PHP 6 будет первой версией с полноценной поддержкой Unicode. Сейчас полным ходом идёт работа по ее реализации во всех стандартных функциях. Внутри будет использоваться кодировка UTF-16, набор доступных символов в которой достаточен, чтобы писать на всех языках мира, живых и мёртвых: там могут оказаться даже такие фантастические языки, как клингон. UTF-16 используется в Java, .NET, Windows и других местах, так что для PHP это хороший выбор.

Проблема состоит в том, что добавление поддержки строк Unicode приводит к замедлению работы ваших сценариев. Конечно, если вы используете китайский или корейский язык, то у вас нет выбора, но тем, кому хватает ASCII, функции работы с многобайтовыми строками – это лишь в три раза более медленное решение, чем обычные. Так что если бы PHP 6 перевёл на UTF-16 всех, web-серверам пришлось бы пойти на вынужденное обновление аппаратного обеспечения одновременно с обновлением версии PHP, просто для того, чтобы обслуживать то же самое число страниц, что и раньше. Решение состоит в том, что в **php.ini** появилась опция, позволяющая прямо в процессе работы включать и выключать поддержку Unicode. На практике это означает, что провайдеры, скорее всего, отключат для вас поддержку многобайтовых строк, и вам придётся специально просить включить её.

Последнее прости

Изменение первой цифры версии – ничто, если оно не может нарушить обратную совместимость. Так что нет ничего удивительного в том, что некоторые хорошо известные возможности обречены на исчезновение из PHP 6. Первым будет отсечён, как и ожидалось, **register_globals**. Включенный по умолчанию в дни старого доброго PHP 4, он стал отключаться, когда появились суперглобальные массивы. Сейчас эта опция будет полностью удалена, и если вы попробуете её включить, то получите ошибку **E_CORE_ERROR** и ссылку на страницу документации. Вам стоило перестать использовать **register_globals** много-много лет назад, но если вы до сих пор не сделали этого, то теперь время вышло.

Что более удивительно, магические кавычки (**magic_quotes**) тоже умрут. Их проблема состоит в нарушении переносимости сценариев – одни люди считают **magic_quotes** полезными, другие – нет, так что в итоге вы никогда не можете предсказать, включено ли на сервере их

использование. Решили эту проблему просто – опция **magic_quotes** полностью удалена, так что всем придётся самостоятельно расставлять слэши в своих переменных. Попытка включить **magic_quotes** в будущем тоже будет приводить к ошибке **E_CORE_ERROR**.

Но это ещё не всё. Мы попрощаемся с safe mode (давайте признаем, никто в здравом уме этот режим не использовал), с режимом совместимости с Zend Engine 1 (это была хорошая штука, но методы клонирования объектов в ZE2 гораздо полезнее), прольём скупую слезу по старым переменным **$HTTP_*_VARS** (с нами остаются наши друзья **$_GET, $POST** и так далее), и нежно простимся с функцией **dl()**. Да, динамическая загрузка расширений умрёт. Правда, это верно только для web, если вы используете CLI SAPI, то **dl()** вам по-прежнему будет доступна. По некоторым вещам мы даже не будем скучать – например, по <%, переходу в режим сценария в стиле ASP или по старой библиотеке для работы с графикой GD1 (просто возьмите вместо неё GD2).

Новые забавы

Из PHP 6 убрали многое, но кое-что при этом добавили. Призовые места среди новых возможностей занимают поддержка goto, функция **ifsetor()** и подсказки типов параметров и возвращаемых значений.

Третье место занимает поддержка goto. Нас всех учили в школе, что пользоваться goto – это плохо. Но ведь конструкция goto не есть абсолютное зло, и при умелом использовании она может сделать ваш код понятнее. Примерно два года назад был предложен патч, который добавлял в PHP поддержку goto, но его сразу же отклонили. Новое решение состоит в том, чтобы расширить функциональность ключевого слова break так, чтобы оно могло выполнять функции goto с тем исключением, что разрешаются только переходы назад. Главной проблемой goto было то, что он легко мог заплести код в спагетти – вы прыгаете вперёд, потом назад, прыжок за прыжком и так далее. Теперь, когда PHP ограничил возможности goto, вам придётся стать действительно некомпетентным программистом, чтобы использовать его неправильно.

Новая функция **ifsetor()** разработана для того, чтобы избежать нудного кода похожего на вот этот:

```
if (isset($_POST["MyVar"])) {
  $myvar = $_POST[ "MyVar"];
} else {
  $myvar =  "wombat";
}
```

С ее помощью вы можете то же самое записать гораздо короче:

```
$myvar = ifsetor($_POST["MyVar"], "wombat");
```

Как вы видите, **ifsetor()** проверяет, установлен ли её первый параметр, и если да, то возвращает его значение. В противном случае она возвращает второй параметр.

Подсказки типов параметров функций нужны для того, чтобы контролировать, что в функцию были переданы параметры нужного типа. Практически они работают как вызов instanceof в первой строке функции. Например:

```
<?php
  class Cat {
    public $Name;
    public function __construct($name) {
      $this->Name = $name;
    }
  }
  class Dog { }
  function meow(Cat $cat) {
    echo "$cat->Name says 'meow!'\n";
  }
  $toby = new Cat( "Toby");
  meow($toby);
?>
```

Использование параметра **Cat $cat** вместо просто **$cat** требует от сценария передавать только объекты типа **Cat** в качестве параметра функции. Если вы попробуете изменить **Cat $cat** на **Dog $cat** (требуя, чтобы передавался объект типа **Dog**), то получите ошибку. Самостоятельно такую проверку можно сделать с помощью **instanceof** следующим образом:

```
function meow($cat) {
  if ($cat instanceof Cat) {
    echo "$cat->Name says 'meow!'\n";
  } else {
    echo  "That's not a cat!\n";
  }
}
```

Подсказки типов возвращаемых значений делают функции PHP гораздо более C-подобными. Убедитесь сами:

```
function Cat Meow(Cat $cat) {
  /// code here
}
```

В этом коде используется тип класса между ключевым словом и именем функции, и если код попытается вернуть из этой функции переменную иного типа, «зажжется» ошибка.

Под капотом

В последней версии PHP было сделано три «незаметных» изменения, которые тем не менее стоит иметь в виду всем программистам.

Во-первых, путаница с синтаксисом [ ] и { } увеличилась. В старые добрые дни [ ] использовали и для того, чтобы индексировать массивы, и для обращения к отдельным символам в строке. Но затем, «чтобы избежать путаницы», было предложено решение использовать [ ] для элементов массива и { } для отдельных символов из строки, вот так: **$myarray[4]** и **$mystring{4}**. Чтобы поддержать это изменение, индексирование строк

> «РАЗРАБОТЧИКИ PHP РЕШИЛИ, ЧТО ИМ ВСЕГДА БОЛЬШЕ НРАВИЛИСЬ []»

через [ ] было объявлено как устаревшее в PHP 4. А теперь, после того, как мы переписали весь наш код на обращение к строкам через {}, разработчики PHP решили, что им всё-таки больше нравится [ ]. И они снова объявили его основным способом, а устаревшим стал {}. Так что конструкция **$mystr[ ]** раньше была неправильной, а теперь стала корректной, тогда как использование в PHP 6 **$mystr{}**, наоборот, будет ошибкой.

Ещё одной причиной для путаницы станет добавление 64-битного целого типа. Сейчас все целые занимают 32 бита в 32-битной системе и 64 бита на 64-битной. И эта разница (а также невозможность работать с очень большими числами на 32-битной платформе) превращается в проблему. В качестве решения было выбрано создание нового типа данных – int64. Обычные целые числа сохранят переменную битность, а вот int64 на любой платформе будет 64-битным.

Наконец, последнее изменение (а его действительно заждались) – стандартная поставка PHP теперь будет включать в себя кэширование скомпилированного кода. Если вы когда-нибудь использовали любую версию Zend Accelerator, то вы представляете, какое серьёзное ускорение он даёт. Alternative PHP Cache (APC) уже некоторое время является частью библиотеки PECL, но включение его в стандартную поставку означает, что настроить его будет гораздо проще. К сожалению, по умолчанию APC будет выключен, поскольку требует определенной настройки, но его присутствие – это уже большой шаг вперёд.

Конечно, на этих двух страницах я мог только слегка коснуться заявленной темы. Но я надеюсь, что я смог дать вам понять, в каких областях у ваших сценариев могут возникнуть проблемы при переходе на следующую версию PHP. PHP 6 уже доступен через CVS и довольно часто обновляется. Чтобы попробовать его, запустите

```
cvs -d :pserver:cvsread@cvs.php.net:/repository checkout php-src
```

и установите у себя. Но не забывайте – всё, о чём мы говорили (и всё, что вы можете скачать через CVS) может измениться к финальному релизу PHP 6. LXF

ЧЕРЕЗ МЕСЯЦ

Мы погрузимся в мир расширений, чтобы узнать, какие из заготовок PHP-кода могут пригодиться в повседневной работе.



ПРОГРАММИРОВАНИЕ СЦЕНАРИЕВ

Уроки Python

ЧАСТЬ 5 Мы долго изучали различные компоненты *Pyhton*, и теперь пришло время сделать что-то по-настоящему полезное. Сегодня мы вместе с **Сергеем Супруновым** перейдем к одному очень важному с практической точки зрения применению языка *Python* – обработке текста.

МЕСЯЦ НАЗАД

Мы изучали классы и объекты Python.

Пожалуй, для языков программирования сценариев типа Perl и *Python* обработка строк является одним из основных применений. Выбрать статистику из log-файла, разобрать «по косточкам» пользовательский ввод, проанализировать содержащиеся в таблице dbf данные… В общем, задачи, как полезные, так и не очень, можно придумывать бесконечно.

Так уж исторически сложилось, что «мы говорим: обработка текста – подразумеваем Perl, мы говорим Perl – подразумеваем: обработка текста». По крайней мере, это любят повторять бородатые администраторы. Сегодня я постараюсь показать, что и *Python* справляется с этим ничуть не хуже, сохраняя при этом все свои преимущества.

Строка как объект

Начнём с более подробного рассмотрения строки как объекта, которому присущи определённые методы. С некоторыми операциями мы уже сталкивались, рассматривая типы данных. Раньше работа со строками велась преимущественно силами модуля **string**. Например, разбиение строки на слова (используя в качестве разделителя пробел) выполнялось примерно так:

```
>>> import string
>>> str = "Строка символов как набор слов"
>>> for s in string.split(str, " "): print s
...
Строка
символов
как
набор
слов
```

Сейчас этот модуль ещё существует, и вы вполне можете его использовать. Однако предпочтительным является использование методов строк. В этом случае каждая строка рассматривается как объект, к которому можно применять те или иные методы. Например, приведённую выше задачу лучше решать так:

```
>>> str = "Строка символов как набор слов"
>>> for s in str.split(" "): print s
...
Строка
символов
как
набор
слов
```

Как видите, намного удобнее. Кстати, вы уже поняли, как работает метод **split()**? Всё правильно – принимая в качестве параметра строку-разделитель (в нашем случае это пробел), он возвращает список под-

строк, разделённых разделителем (извините за каламбур). Сам разделитель в итоговые подстроки не входит. Мы могли вообще не указывать параметр, поскольку пробел используется по умолчанию: **str.split()** вернёт тот же результат.

Рассмотрим ещё несколько примеров:

```
>>> a = 'just a test line'
>>> a.count('t')
3
>>> a.center(25)
' just a test line '
>>> a.rjust(30)
' just a test line'
>>> '-'.join(a.split())
'just-a-test-line'
>>> a.title()
'Just A Test Line'
>>> a.title().swapcase()
'jUST a tEST lINE'
```

Ещё не забыли, как получить полный список доступных методов? Всё правильно – наша любимая функция: **dir('qwe')**.

Входит и выходит

Говоря о строках, коротко остановимся на вводе данных и форматировании вывода. И с тем, и с другим мы уже сталкивались, но не вдавались в подробности.

Итак, за пользовательский ввод отвечают две функции: **input()** и **raw_input()**. В качестве аргумента они принимают строку-приглашение (она может быть пустой). Отличаются тем, что первая требует форматированного ввода – строки должны заключаться в кавычки, числа без кавычек воспринимаются именно как числовые типы данных (в зависимости от формата), и т.д. Неправильный ввод, например, отсутствие кавычек в строке, будет приводить к ошибке:

```
>>> a = input("Введите число: ")
Введите число: 5
>>> print a
5
>>> a = input("Введите число: ")
Введите число: число
Traceback (most recent call last):
 File "<stdin>", line 1, in ?
 File "<string>", line 1
 число
 ^
SyntaxError: invalid syntax
```

В отличие от этого, вторая функция – **raw_input()** – воспринимает весь ввод в «сыром» виде как обычную строку, не пытаясь разобраться в принадлежности к конкретному типу данных. Эта задача возлагается на программиста.

Теперь – пара слов о форматированном выводе. Использование знакомест **%s** – это его зачатки. На самом деле, возможности данного механизма гораздо богаче:

```
>>> a = 0
>>> b = "один"
>>> c = 2.34
>>> print "Значения: %d, %s, %f" % (a, b, c)
Значения: 0, один, 2.340000
>>> print "Три знака после запятой: %0.3f" % c
Три знака после запятой: 2.340
>>> print "Выравнивание вправо: %20s" % b
Выравнивание вправо: один
```

Можно также использовать так называемый «словарный» синтаксис. В качестве примера рассмотрим небольшой сценарий, анализирующий учётные записи пользователей:

```
#!/usr/bin/Python
fd = open('/etc/passwd', 'r')
line = 'none'
user = {}
while line != '':
 line = fd.readline()
 if line != '' and line[0] != '#':
 tmp = line.split(':')
 user['login'] = tmp[0]
 user['uid'] = tmp[2]
 user['homedir'] = tmp[5]
 print "| %(login)-5s | %(uid)5s | %(homedir)-14s |" % user
fd.close()
```

Как видите, можно создавать именованные знакоместа вида **%(key)s**, где **key** – ключ словаря, который передаётся оператору **print**. Есть один интересный приём – вы всегда можете воспользоваться функцией **vars()**, которая вернёт словарь инициированных в данный момент переменных. То есть к переменной **qwe** вы можете обратиться и так: **vars()['qwe']**. Как следствие, вставить значение переменной **qwe** в вывод оператора print можно таким образом: **print "%(qwe)s" % vars()**.

Знак «минус» перед числом, указывающим ширину текстового поля, означает выравнивание по левому краю (по умолчанию используется выравнивание вправо).

Результат работы приведённого выше сценария будет выглядеть примерно так:

```
serg$ ./py3.py
| root | 0 | /root |
| www | 80 | /nonexistent |
| ftp | 21 | /home/ftp |
| oops | 5002 | /nonexistent |
| dspam | 8080 | /var/db/dspam |
```

Развивая наш сценарий, можно добавить «шапку» таблицы, и т.д. Оставлю это вам в качестве упражнения.

Регулируем выражения

Безусловно, никакие методы строк не сравнятся по своей мощи с регулярными выражениями. В языке *Python* за работу с ними отвечает модуль re (когда-то это был самостоятельный модуль, сейчас – просто «обёртка» к более мощному модулю sre).

Говоря простым языком, регулярное выражение (РВ) – это специальная строка, описывающая группу строк, имеющих определённый формат. По сути, РВ – это шаблон, которому может соответствовать несколько различных строк. Например, выражение **«[a-z]»** описывает строки, состоящие из одной строчной буквы латинского алфавита. Специальные символы, использующиеся в регулярных выражениях, можно разделить на три группы: описатели, квантификаторы и символы группировки.

Описатели – это выражения, описывающие определённый набор символов, которые могут им соответствовать. Например, **«[0-9]»** описывает любую цифру, **«\s»** – любой пробельный символ, **«.»** – любой символ. Просто символ, не принадлежащий специальным или экранированный обратным слэшем (например, **«m»**, **«\[»**) описывает сам себя.

Символы группировки – круглые скобки – позволяют объединить несколько описателей в группу, которая будет рассматриваться как единое целое.

Наконец, квантификаторы задают число повторений символов, соответствующих описателю или группе, после которых он задан. Например, **«.*»** означает любое (в том числе, нулевое) количество любых символов, **«\d+»** требует наличия как минимум одной цифры, выражению **«([a-z][0-9]){2,4}»** будет соответствовать строка, в которой группа из буквы и цифры повторяется от двух до четырёх раз – **f5e3, o2o2o2o2**.

Нужно помнить об одном свойстве квантификаторов, именуемом «жадностью» – каждый квантификатор стремится захватить как можно больше символов, соответствующих описателю. Своим же «коллегам», стоящим после них, они оставляют минимум, необходимый для того, чтобы строка считалась соответствующей выражению. Например, выражение **«.+.+»** требует, чтобы в строке было две последовательности любых символов, как минимум с одним символом в каждой. Строка

НЕКОТОРЫЕ ОПИСАТЕЛИ

\A – начало строки
\Z – конец строки
\b – граница слова
\d – любая цифра
\s – пробельный символ
\w – цифра или буква

>>

НЕ ВСЕ ТАК ПРОСТО

Стоит отметить, что вполне рядовая на первый взгляд задача поиска адресов e-mail на самом деле очень и очень сложна. Стандарт RFC822, определяющий формат адреса электронной почты, предусматривает произвольную глубину вложенности комментариев (которые заключаются в круглые скобки) – подобные конструкции регулярным выражениям просто не по зубам. Полный код регулярного выражения, ограниченного одним-единственным уровнем вложенности комментариев, содержит 4724 символа! Подробности можно найти в замечательной книге Дж. Фридла «Регулярные выражения. Библиотека программиста», выпущенной издательством O'Reilly и переведенной на русский язык издательством «Питер».

<< **«qwerty»** будет ему соответствовать, при этом первый **«+»** поглотит все символы, кроме последней буквы, которую необходимо отдать второму квантификатору.

Жадность квантификатора можно поубавить, поставив после него символ **«?»**. Например, **«.+?.+?»** будет вести себя уже по-другому – первый **«+»** заберёт себе один символ (по принципу, «а мне больше и не надо»), а всё остальное придётся «кушать» второму квантификатору.

Казалось бы, не всё ли равно, кто сколько символов захватит? Главное ведь чтобы результат был одинаковый? Если нам нужно просто определить соответствие, то, конечно, разницы никакой. Но если мы используем группировку для последующей обработки фрагментов (см. пример ниже), то о «жадности» забывать нельзя.

Однако хватит теории (о регулярных выражениях много и хорошо написано в различных книгах, статьях и на man-страницах). Вернёмся к модулю re.

Наиболее типичный способ его использования – создание объекта-шаблона с помощью метода **re.compile()** с последующим его применением к конкретным строкам с помощью методов **re.match()**, **re.search()**, **re.sub()** и т.д. Методы **match()** и **search()** отличаются тем, что первый требует соответствия РВ с начала строки, а второй проверяет наличие подходящего фрагмента в любом месте строки:

```
>>> a = re.compile('\d+')
>>> print a.search('qwe123') or "Нет соответствия"
<_sre.SRE_Match object at 0x81bbd78>
>>> print a.match('qwe123') or "Нет соответствия"
Нет соответствия
>>> print a.match('123qwe') or "Нет соответствия"
<_sre.SRE_Match object at 0x81bbd78>
```

Как видите, в случае соответствия как **match()**, так и **search()** возвращают объект класса **SRE_Match**. Получить фрагмент, удовлетворяющий РВ, можно с помощью метода **group()** этого объекта:

```
>>> print a.search('qwe123qwe').group()
123
```

В данном примере мы «вырезали» число из строки (чему при желании можно найти достаточно много практических применений).

Метод **sub(текст_замены, строка)** выполняет замену соответствующих шаблону фрагментов заданной строкой:

```
>>> a = re.compile('\d+')
>>> print a.sub('несколько', 'Стоит 50 рублей')
Стоит несколько рублей
```

Пример использования ещё одного полезного метода – **findall()** – вы найдёте в рассматриваемом ниже сценарии. Ну а в том, как работает **split()**, думаю, разберётесь сами.

В качестве примера решим задачу – выбрать из некоторого текста все адреса электронной почты (только не подумайте, что я помогаю спамерам – скальпель тоже иногда может пользу приносить). Чтобы не сильно усложнять алгоритм, будем считать, что фрагмент строки является электронным адресом, если он соответствует следующим условиям:

- обязательное наличие символа «@» внутри фрагмента;
- до «@» может стоять любое число латинских букв, цифр и символов «точка»;
- после «@» должно находиться не менее двух групп латинских букв, цифр и дефисов, разделённых точками; причём последняя такая группа может содержать от двух до четырёх букв (другие символы не допускаются).
- к регистру символов придираться не будем.

Пример «правильного» фрагмента: Vasya.Pupkin@gdeto-tam.D12.info. Записать наши требования можно с помощью такого выражения:

```
([\w.]+@([\w-]+\.)+[a-z]{2,4})\W
```

Завершающий определитель **\W** нужен для того, чтобы после последней группы обязательно стоял не цифро-буквенный символ (в противном случае qwe@qwe.qwerty будет соответствовать РВ, т.к. квантификатор возьмёт себе необходимые 4 символа, а остальное просто не будет приниматься во внимание). Нужный же нам адрес будет формироваться в первой группе (внешние скобки).

Внимательные читатели, думаю, уже заметили, что нашему шаблону будет соответствовать и такой **«адрес»: ...@-----.-.ru**. Небольшим усложнением нашего «фильтра» эту проблему можно решить – попробуйте на досуге.

Базовая версия нашего сценария выглядит следующим образом:

```
#!/usr/bin/Python
# -*- coding: koi8-r -*-
import re
def getmail(text):
 filter = re.compile(r'([\w.]+@([\w-]+\.)+[a-z]{2,4})\W', re.I)
 for i in filter.findall(text):
 print i[0]
example = '''\
```

Этот текст служит для проверки работы функции **getmail**.

Он должен нормально распознавать такие адреса как v**asya@mail.ru**, **Vasya.Pupkin@gde-to.tam.D12.info, u_12@_test.me** и **12.34@56-78.90.ua**.

Адреса вида **@mail.ru**, **mail@mail** и **qwe@domain.qwerty** и **qwe@ma.ru123** должны игнорироваться.

К сожалению, здесь не решена проблема адресов вида: **...@---.--.ru**. В принципе, исправить это не сложно.

```
'''
getmail(example)
```

Результат его работы:

```
serg$ ./getmail.py
vasya@mail.ru
Vasya.Pupkin@gde-to.tam.D12.info
u_12@_test.me
12.34@56-78.90.ua
...@---.--.ru
```

Вторым параметром метода **compile()** задаются флаги, в частности, флаг **re.I (re.IGNORECASE)** отключает проверку регистра символов. Дополнительных пояснений, думаю, не требуется. Замечу лишь, что **findall()** возвращает в кортеже все найденные группы в порядке обнаружения первой скобки. Таким образом, первым элементом и будет искомый адрес электронной почты. Ещё обратите внимание на то, что перед строкой регулярного выражения стоит буква **«r»**. Она говорит о том, что последующую строку следует рассматривать как «сырую», то есть не обрабатывать специальные символы, экранированные обратным слэшем.

Что ж, большая часть нашего пути позади. Осталось рассмотреть последний крупный вопрос: разработку приложений с графическим интерфейсом, чем в следующий раз и займёмся. **LXF**

НЕКОТОРЫЕ КВАНТИФИКАТОРЫ

***** – любое количество фрагментов
+ – один и более фрагментов
? – ноль или один фрагмент
{m,n} – не меньше **m** и не больше **n**

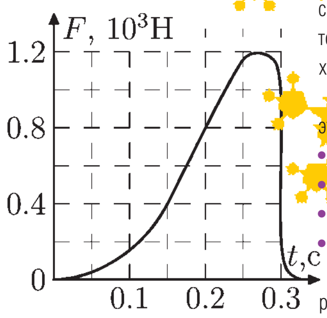
ЧЕРЕЗ МЕСЯЦ

Мы рассмотрим создание приложений с графическим интерфейсом пользователя.

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

MetaPost Графики и диаграммы

ЧАСТЬ 4 Даже жизнь можно отобразить в виде графика функции от времени. **Евгений Балдин** расскажет, как выразить это на языке Metapost.



График, созданный вручную

Отображение данных на бумаге всегда было и останется нетривиальным процессом. Несмотря на то, что получение данных, как правило, занимает гораздо больше ресурсов, задача оформления графиков достойна автоматизации. В частности, автоматизация позволяет один раз придумать, как представить данные наилучшим образом, и в дальнейшем не снижать уровень оформления.

По-хорошему, графики следует создавать в специализированных приложениях. Автоматизация этого процесса достаточно специфична, чтобы реализовывать его с помощью программ, не предназначенных только для этого. Если вам нужны двумерные графики, то никто не справится с этим лучше чем gnuplot. Если вас интересуют гистограммы, то это работа для пакетов анализа **paw/cernlib** или **root**. Но если вам хочется разукрасить график, то *MetaPost* будет для этого очень кстати.

Ничего не мешает нарисовать график самостоятельно. Всё, что для этого нужно:

- знать диапазон функции и её аргумента,
- нарисовать оси координат,
- создать сетку и поставить метки на осях,
- изобразить функцию.

В этом смысле кодирование графика ничем не отличается от кодирования рисунка – нужно лишь уметь рисовать стрелки и линии. Однако это годится только для простых графиков.

Графики в MetaPost

Для серьёзной работы с графиками в *MetaPost* есть довольно продвинутый пакет **graph.mp**. Этот пакет написан «отцом» *MetaPost* Джоном Хобби, и к нему прилагается подробная документация, которую можно найти в стандартной поставке *LaTeX* в виде файла **mpgraph.pdf**. Если вы планируете воспользоваться этим пакетом, то прежде изучите этот текст.

```
%Файл graphics.mp
input graph;
%График переменной скорости - тема "дифференцирование"
beginfig(2) ;
 numeric u;
 u = 0.6mm;
 %начало графика
 draw begingraph(12cm,3cm);
  path r;
  numeric e,pi,A,sigma,n,scale,mean,y[],x[];
  e:=2.718;pi:=3.14159;A=50;
  sigma=1.4;n:=100;scale=10;mean=5;
  %создание функции - она использовалась не один раз
  for j:=0 upto n:
   if (j=0):
             x[0]=0;y[0]=0;
   else:
             x[j]:=scale*j/n;
             y[j]:=y[j-1]+A*(e**(-(((scale*j/n-mean)/
    (1.41*sigma))**2)))*
             (x[j]-x[j-1])/(sqrt(2*pi)*sigma);
   fi
```

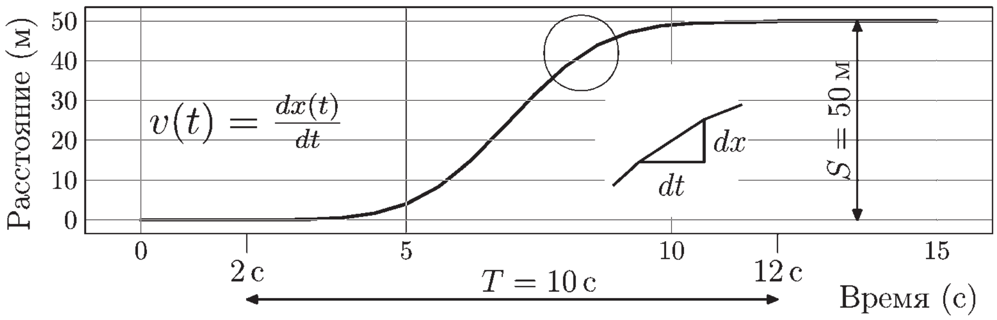


Иллюстрация для объяснения понятия дифференцирования

<<

```
endfor;
%здесь нужна функция из отрезков, поэтому step 6
r:=(0,0)for j:=0 step 6 until (n-1):
  --((2,0)+(x[j],y[j])) endfor ..(12,50)--(15,50);
%отрисовка функции
gdraw r withpen pencircle scaled 0.8u;
%сетка
autogrid(grid.bot,grid.lft) withcolor .5white;
%дополнительные метки на оси абсцисс
otick.bot(btex 2\,\text{c} etex,2);
otick.bot(btex 12\,\text{c} etex,12);
%текст и стрелка за пределами графика
gdrawdblarrow (2,-20)--(12,-20)
      withpen pencircle scaled 0.5u;
glabel.top(btex \(T=10\,\text{c}\) etex,(7.5,-20));
gdrawdblarrow (13.5,0)--(13.5,50)
      withpen pencircle scaled 0.5u;
%текстовые метки внутри графика
glabel.lft(btex \(S=50\,\text{м}\) etex
    rotated 90, 1/2[(13.5,0),(13.5,50)]);
glabel.rt(btex \(v(t)=\frac{dx(t)}{dt}\) etex
        scaled 1.3,(0,25));
glabel(btex \(dx\) etex,(11.1,20)) ;
glabel(btex \(dt\) etex,(10,9)) ;
%подписи к осям
glabel.rt(btex Время (c) etex,(13,-20)) ;
glabel.lft(btex Расстояние (м) etex
    rotated 90,OUT) shifted (0cm,0.5cm);
endgraph;
endfig ;
```

В **graph.mp** определено специальное окружение. Между **begingraph** и **endgraph** действует своя система координат. Она привязана не к геометрическим размерам картинки, а к диапазону осей графика. То есть точка (0,0) соответствует точке пересечения оси абсцисс и оси ординат графика. Размер создаваемого графика указывается сразу после **begingraph**.

Вместо стандартных команд draw, fill, label и dotlabel используются gdraw, gfill, glabel и gdotlabel, соответственно. Новые команды работают с учётом координат графика. С их помощью можно построить функцию именно по точкам, не заботясь о сдвигах и тому подобном. Для подписи осей существует специальная точка OUT, которая в зависимости от суффикса команды glabel выносит текст за пределы основной сетки.

Функция «отрисовывается» точно так же, как любая из кривых: создаётся путь, который затем выводится с помощью команды gdraw.

Для фиксации диапазона графика используется команда:

```
setrange("нижний левый угол","верхний правый угол");
```

При вызове этой инструкции все последующие функции работают в указанном диапазоне. Таким образом, используя одно и то же пространство, можно совместить несколько графиков, имеющих различный диапазон аргументов и функций. Шкалу графика можно сделать логарифмической с помощью инструкции setcoord:

```
%x- линейная шкала, y - логарифмическая
setcoord(linear,log);
```

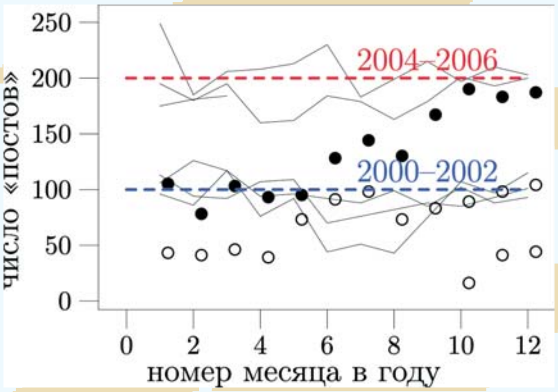
Для создания сетки и меток используется команда autogrid:

```
%сетка по нижней и левой осям
autogrid(grid.bot,grid.lft) withcolor .5white;
%внешние метки по нижней оси и внутренние метки по правой оси
autogrid(otick.bot,itick.rt);
```

Работа с файлами данных

С помощью команды gdraw можно читать данные из файла. Если в качестве аргумента передаётся текстовая строка, то gdraw предполагает, что это имя файла.

```
%Файл graphics.mp
%Число постов на LOR от месяца года
```



Число постов на linux.org.ru

```
beginfig(4) ;
 u := 0.4mm;
 draw begingraph(150u,100u);
  pickup pencircle scaled 0.2u;
  gdraw("lor-1998.dat") plot btex \(\circ\) etex;
  gdraw("lor-1999.dat") plot btex \(\circ\) etex;
  gdraw("lor-2000.dat");
  gdraw("lor-2001.dat");
  gdraw("lor-2002.dat");
  gdraw("lor-2003.dat") plot btex \(\bullet\) etex;
  gdraw("lor-2004.dat");
  gdraw("lor-2005.dat");
  gdraw("lor-2006.dat");
  gdraw (0,100)--(12,100) withpen pencircle scaled 1u
        dashed evenly scaled 1u withcolor blue;
  glabel.top(btex 2000--2002 etex,(9,100)) withcolor blue;
  gdraw (0,200)--(12,200) withpen pencircle scaled 1u
  dashed evenly scaled 1u withcolor red;
  glabel.top(btex 2004--2006 etex,(9,200)) withcolor red;
  glabel.lft(btex число «постов» etex rotated 90,OUT);
  glabel.bot(btex номер месяца в году etex,OUT);
 endgraph;
endfig;
```

Как и команда draw, gdraw «понимает» инструкции withpen (какое перо использовать), withcolor (цвет линии) и dashed (использовать пунктир). Дополнительно gdraw воспринимает команду plot {picture}. Указанная в инструкции plot картинка выбирается в качестве маркера. На рисунке чёрными маркерами отмечен 2003 год – год, когда произошло удвоение ежемесячного числа новостных постов на LOR (http://www.linux.org.ru), а белыми маркерами – первый год существования этого ресурса.

По умолчанию предполагается, что входной файл представляет из себя два столбца цифр, разделённых пробелами. Пример файла **lor-2006.dat (**данные взяты с сайта http://www.linux.org.ru):

```
1 175
2 181
3 184
```

Если данные имеют более сложное представление, чем в упомянутом выше файле, то используется команда gdata.

```
gdata («имя файла», переменная куда считываются данные, действие);
```

Если на текстовый файл вида:

```
1 7.289 H
2 2.425 He
--вырезано--
26 -57.710 Mn
27 -60.604 Fe
```

воздействовать с помощью следующего кода:

```
%Файл graphics.mp
%дефект масс
beginfig(5) ;
 u := 0.8mm;
```

```
draw begingraph(150u,100u);
 gdraw “mendeleev.dat” dashed evenly scaled 1u
                   withcolor 0.5white ;
 gdata(“mendeleev.dat”, s,
  glabel(s3,(scantokens s1,scantokens s2));
  )
 glabel.lft(btex Дефект массы (МэВ) etex rotated 90,OUT);
 glabel.bot(btex Порядковый номер элемента в таблице etex,OUT);
endgraph;
endfig;
```

то получится простенькая картинка:



Дефект массы химических элементов от порядкового номера в периодической таблице Меделеева. Приведены элементы от водорода до железа.

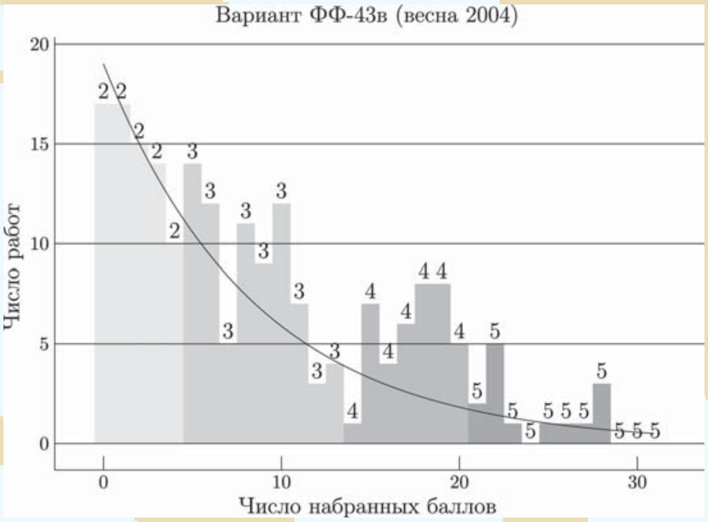
Переменная **s** объявляется массивом, в который считывается строка. В качестве разделителя выступает пробел. Запись **s1** соответствует **s[1]** – первому элементу массива. Макрос scantokens «встраивает» значение аргумента в код. В данном случае происходит перевод строки в число. В качестве «действия» в команде gdata может быть набор из нескольких команд.

Гистограммы в MetaPost

Если вы хотите нарисовать гистограмму, то можно воспользоваться следующими макросами:

```
%Файл graphics.mp
%создает профиль гистограммы из пути
def histpath(expr pairs) =
 for i=0 upto length pairs:
  if (i>0):--else:fi((point i of pairs)-
   ((xpart(point 1 of pairs)-
    xpart(point 0 of pairs))/2,0))--
  ((point i of pairs)+
   ((xpart(point 1 of pairs)-
    xpart(point 0 of pairs))/2,0))
 endfor
enddef;
%создает замкнутый профиль гистограммы
%для закрашивания
def histpathcycle(expr pairs) =
 ((xpart(point 0 of pairs),0)-
  ((xpart(point 1 of pairs)-
   xpart(point 0 of pairs))/2,0))
 --histpath(pairs)--
 ((xpart(point infinity of pairs),0)+
  ((xpart(point 1 of pairs)-
   xpart(point 0 of pairs))/2,0))--cycle
enddef;
```

Распределение, представленное на картинке, похоже на убывающую экспоненту. Вот так рисовались «пятёрки»:



Распределение по баллам, полученным на Открытой олимпиаде по физике в НГУ в 2004 году.

```
%Файл graphics.mp
%Оценки
path five;
five:=(21,2)--(22,5)--(23,1)--(24,0)--(25,1)--
   (26,1)--(27,1)--(28,3)--(29,0)--(30,0)--(31,0);
%Заполняем гистограмму
gfill histpathcycle(five) withcolor 0.6white;
```

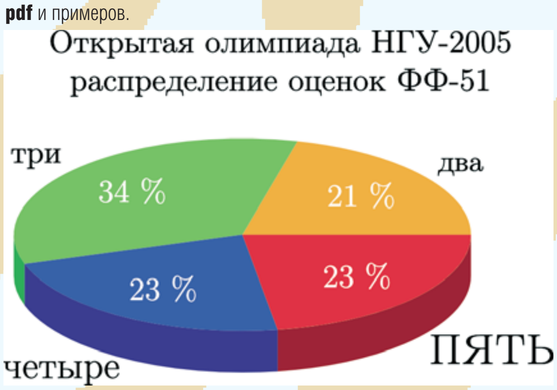
Для закрашивания контура применяется инструкция gfill – аналог команды fill.

Данные по оценкам получались в результате обработки текстовых файлов со статистикой с помощью простого скрипта на Perl. Время, которое потребовалось на автоматизацию, составило примерно один человеко-день, что многократно меньше времени написания полного отчёта, куда вошли подобные гистограммы, и оформления всех условий и решений Открытой олимпиады и последовавших за ней летних вступительных экзаменов.

Создание своих макросов на *MetaPost* является стандартным действием. META не страдает избыточностью и поощряет «доводку» окружения под ваши нужды.

Круговые диаграммы

Пакет **piechartmp.mp** относительно молод. Замечательная, можно сказать красочная, документация поставляется в виде файла **piechartmp.pdf** и примеров.



Распределение оценок.

Как обычно, за русским языком надо приглядывать. По умолчанию пакет работает только с латиницей.

```
%Файл pie.mp
input piechartmp;
%Круговые гистограммы
%Открытая Олимпиада “ФФ-51”
beginfig(1) ;
 numeric u; u:=1mm;
```

>>

```
%чтобы был русский
SetupText(1, "\input{preheader-base}", "\begin{document}")
label(btex Открытая олимпиада НГУ-2005 etex,(5u,25u));
label(btex распределение оценок ФФ-51 etex,(5u,20u));
%Чтобы отображался процент
SetupPercent(this, " \%");
%Опеределение сегментов
Segment( 15,"\small два", auto);
Segment( 24,"три", auto);
Segment( 16,"\large четыре", auto);
Segment( 16,"\huge пять", red );
%Создание круговой гистограммы
PieChart(30u, 0.2, 65, 0, 0);
%метки
Label(0)(percent)(inwards,0) withcolor white;
Label.auto(0)(name)(outwards,0);
endfig;
```

Алгоритм создания круговой или секторной диаграммы следующий:

- Если вы хотите воспользоваться автоматической системой размещения меток, то с помощью команды SetupText следует настроить вывод текста. В файле **preheader-base.tex** должна быть минимальная шапка для документа *LaTeX*. Если вы хотите использовать русский язык, то там обязательно должны быть строки вида **\usepackage[T2A]{fontenc}** и **\usepackage[koi8-r]{inputenc}**. Вместо koi8-r можно поставить свою кодировку.
- С помощью команды Segment следует определить сегменты. В качестве аргументов команде передаётся число (измеряемая величина), текстовая метка и цвет сегмента (можно указать значение по умолчанию – auto).

Можно также ввести четвёртый необязательный строковый параметр, который заменяет измеряемую величину при создании меток. Это решает проблему ограничения на диапазон чисел в *MetaPost*.

- Нарисовать гистограмму. В качестве параметров команде PieChart передаётся размер, высота диаграммы, угол, под которым мы на неё смотрим (трёхмерность), угол поворота вокруг центральной оси и «смещение» сегментов относительно центра.
- Расставить метки.

В момент создания каждому сегменту присваивается порядковый номер. Отсчет начинается с единицы. Все изменения в круговой диаграмме делаются глобально. Пакет написан так, чтобы в одном mp-файле было удобно работать ровно с одной диаграммой. Если при описании диаграммы внутри окружения beginfig вы уже что-то определили, то нет необходимости повторять это в последующих окружениях. Этот режим удобен, если нужно создать несколько модификаций одной диаграммы, например, для создания «оверлеев» в презентации. Минусом такого подхода является то, что если в этом же **mp**-файле хочется создать ещё одну гистограмму, то уже определённые сегменты необходимо спрятать.



Распределение «вещества» в современной вселенной.

```
%Круговые гистограммы
%распределение вещества во вселенной
beginfig(2) ;
 numeric u; u:=1mm;
 %прячем определённые ранее сегменты
 SegmentState(1,hidden,this);
 SegmentState(2,hidden,this);
 SegmentState(3,hidden,this);
 SegmentState(4,hidden,this);
 %Опеределение новых сегментов
 Segment( 65,"\Large \bf тёмная энергия",green,"\LARGE 65\%--70\%");
 Segment( 25,"\Large \bf тёмная материя", (1,0,1),"\LARGE 25\%");
 Segment( 0.5,"звёзды", (1,1,0),"\LARGE 0.5\%");
 Segment( 5,"обычное вещество", red,"\LARGE 5\%");
 Segment( 1,"нейтрино", black,"\LARGE 0.3\%--3\%");
 %Выдвигаем сегменты из диаграммы
 SegmentState(7,this,0.7);
 SegmentState(9,this,0.7);
 SegmentState(8,this,0.7);
 %Создание круговой гистограммы
 PieChart(30u, 0.3, 30, 340, 0);
 %метки
 Label.auto(0)(name)(outwards,0) ;
 Label.auto(5)(value)(inwards,0) shifted (7u,-7u) withcolor white;
 Label.auto(6)(value)(inwards,0) shifted (0u,5u) withcolor white;
 Label.auto(7)(value)(inwards,(0u,-15u)) withcolor (1,1,0);
 Label.auto(8)(value)(inwards,(25u,10u)) withcolor red;
 Label.auto(9)(value)(inwards,(20u,20u)) withcolor black;
endfig;
```

Состояние каждого из сегментов можно задавать с помощью команды SegmentState[1].

Состояние сегмента может быть:

- normal – сегмент становится видимым,
- invisible – сегмент не рисуется при создании круговой диаграммы (в диаграмме остаётся пустое место),
- hidden – при создании диаграммы этот сегмент игнорируется.

Радиальный сдвиг сегмента указывается в процентах, где 1 – это 100%, то есть сегмент полностью «выдвинут».

Для установки меток используется команда Label[2]:

Label понимает те же суффиксы, что и обычная команда label, плюс auto. При установке суффикса auto Label пытается сама угадать, где лучше поставить метку.

В качестве порядкового номера сегмента можно передать 0. В этом случае команда Label применяется ко всем видимым сегментам. Label может принимать в качестве аргумента список порядковых номеров сегментов, разделённый запятыми (например: 5, 7, 9).

Метка может быть простой текстовой строкой или одним из стандартных значений:

- value – измеряемая величина, которая приписывается сегменту при его создании,
- percent – процент занимаемой площади,
- name – имя сегмента, которое приписывается ему при его создании.

Базовая точка представляется в виде пары чисел **(x,y)**, где **x** – расстояние от вершины сегмента (**0** – это вершина, **1** – это противоположный от вершины край), а **y** – эквивалент полярного угла (**0** – край сегмента по часовой стрелки, **1** – край сегмента против часовой стрелки). В пакете определены константы **inwards=(0.7,0.7)** и **outwards=(1.1,0.5)**. Сдвиг же представлен в терминах всего графика (например, **(1cm,0)** означает сдвиг на **1** сантиметр вправо).

При желании тип заливки и порядок цветов по умолчанию можно определять самостоятельно. Подробнее об этом рассказано в документации к пакету **piechartmp.pdf**.

[1] SegmentState – порядковый номер сегмента, устанавливаемое состояние сегмента или this если оно не меняется, радиальный сдвиг сегмента или this если он не меняется;

[2] Label – порядковый номер сегмента, метка, базовая точка в системе отсчёта сегмента, сдвиг.

LaTeX рисует с помощью MetaPost

Для рисования графиков можно воспользоваться и возможностями самого *LaTeX*. Например, с этой обязанностью прекрасно справляется стандартный пакет *mfpic*. Для «отрисовки» функций с помощью этого пакета достаточно задать саму функцию, а не рисовать по точкам:

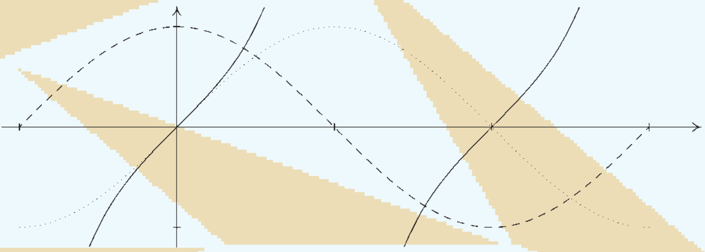


График косинуса, синуса и тангенса

```
%файл graphics-mfpic.tex
%Пример от Сергея В. Знаменского
\documentclass{article}
\usepackage[koi8-r]{inputenc}
\usepackage[russian]{babel}
\usepackage[MetaPost]{mfpic}
%указываем mp-файл куда будет «складываться» код mfpic
\opengraphsfile{graph-mfp}
\begin{document}
%не печатать номер страницы
\pagestyle{empty}
%начинаем создание графика
\mfpic[1][57.2]{-100}{300}{-1.2}{1.2}
%оси координат
\axes
\xmarks{-90,90,180,270}
\ymarks{-1,1}
%рисуем синус точками
\dotted\function{-90,270,4.5}{sind(x)}
%рисуем косинус пунктиром
\dashed\function{-90,270,4.5}{cosd(x)}
%рисуем тангенс сплошной линией
\function{-50,50,10}{tand(x)}
\function{130,230,10}{tand(x)}
%заканчиваем создание графика
\endmfpic
%закрываем mp-файл
\closegraphsfile
\end{document}
```

Для получения картинки необходимо выполнить следующие действия:

```
> LaTeX graphics-mfpic.tex
> mpost graph-mfp.mp
> LaTeX graphics-mfpic.tex
> dvips -E graphics-mfpic.dvi -o graphics-mfpic.eps
```

При компиляции файла graphics-mfpic.tex создаётся graph-mfp.mp, куда пишутся команды на языке META. Если посмотреть на код **graph-mfp.mp**, то можно увидеть, что там используется макрос function, который позволяет рисовать функции одной командой, без всяких циклов. Как это делается, можно подглядеть в библиотечке grafbase.mp, которая входит в состав пакета *mfpic*.

Более подробно про пакет *mfpic* можно узнать из документации которую можно найти в директории **$(TEXMF)/texmf-dist/doc/generic/mfpic/**, где **$(TEXMF)** – директория в которую установлен *LaTeX* (это заведомо верно для дистрибутива *TeX Live*).

gnuplot

В случае отображения больших объёмов данных лучше использовать *gnuplot*. Чтобы *gnuplot* работал с *MetaPost*, необходимо определить правильное устройство вывода. Для этого в *gnuplot* следует передать команду:

```
set terminal mp color LaTeX
```

В этом случае *gnuplot* будет выводить графики в формате *MetaPost*. К сожалению, похоже, что вывод в *MetaPost* давно не поддерживался, поэтому установка кодировки:

```
set encoding koi8r
```

не оказывает должного эффекта. Получившийся mp-файл приходится дорабатывать вручную. К счастью, это не сложно (следует поправить только заголовок), но для автоматизации следует озадачиться исправлением этой неприятной ошибки в *gnuplot*.

В случае, если возникнет желание изменить метки, следует обратить внимание на переменную textmag, которая отвечает за размер текста.



Интегральная светимость от времени. Данные за полтора года (примерно 130 тысяч точек) обработаны с помощью программы *gnuplot*.

Подведём итоги

С помощью *MetaPost* можно создавать двумерные графики и диаграммы любой сложности. Возможности языка META и стандартные пакеты позволяют сконцентрироваться на смысловой части и не отвлекаться на отдельные элементы оформления. LXF

ЧЕРЕЗ МЕСЯЦ

Нас ждет обзор стандартных пакетов и полезных приемов.

Вопрос? Ответ!

Если вы бьетесь над решением какой-то проблемы и HOWTO не помогают, почему бы не написать нам? Наши эксперты ответят даже на самые сложные вопросы.

НАШИ ЭКСПЕРТЫ

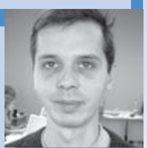
Наши эксперты найдут ответ на самый трудный ваш вопрос. Если у вас проблемы с установкой, настройкой модема, сетью или еще чем-нибудь – просто напишите нам, а обо всем остальном позаботимся мы.



Управляя Интернет-провайдером, а заодно подрабатывая редактором дисков *LXF*, **Нейл Ботвик** (Neil Bothwick) скромно зовет себя мастером на все руки.



Сергей Владимирович Рыжиков
Генеральный директор компании «БИтрикс»



Максим Владимирович Смирнов

Ведущий программист отдела разработки компании «БИтрикс»

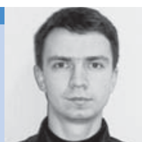


Александр К. – сторонник Unix-way. Молодой, но перспективный член дружной команды экспертов.



Валентин Синицын
Поддерживает проект Slackware Reiser4, интересуется настольными Linux-технологиями и рад помочь Вам разобраться с ними.

Уважаемые читатели!
В апрельском выпуске рубрики была допущена досадная оплошность: нами не был упомянут новый представитель команды экспертов:



Андрей Маркелов
RHCE/RHCI
Руководитель отдела технической поддержки
НОУ «Инвента»

Редакция приносит свои извинения.

Двое на одного

В На моем ноутбуке Dell Precision M60 установлена SUSE 10.0. К сожалению, мне (видимо, как и всем, чьи сайты я смог отыскать через Google) не удалось подключить к нему внешний монитор. Я пытался подсоединять и старый аналоговый монитор Dell, и новейшую ЖК-панель Sony KLV-S23A10 с ПК-разъемом. Пробовал подключать напрямую (без удлинителя) и через аналоговый преобразователь. В ноутбуке для переключения между внешним и внутренним видеовыводом служит комбинация клавиш Fn+F8; у меня это заканчивается зависанием.

Я изучал настройки BIOS (жал F2 при загрузке), попробовал оба источника видео, System и Dock... Следовал различным HOWTO, перекопал свой xorg.conf, добавив новый монитор, Display и Screen... Менял разрешение, частоту вертикальной и горизонтальной синхронизации, чтобы она соответствовала параметрам монитора (1280 768, 47.4 КГц и 60 Гц)... Все впустую. *YaST* не находит никаких новых устройств.

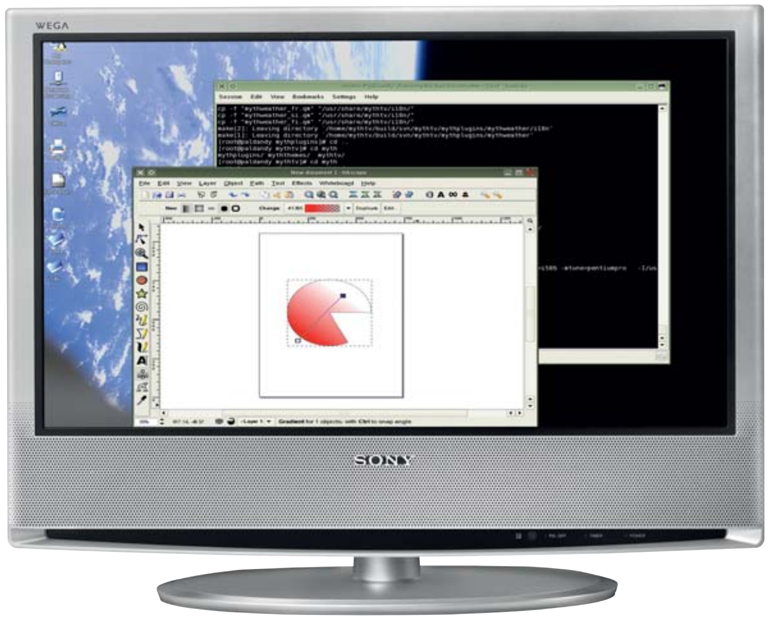
На схожем ноутбуке (Dell Latitude) с Windows XP обнаружение внешнего монитора прошло без проблем, так что я думаю, что проблема в ядре Linux или в его варианте для SUSE.

К. Эндрюс Лаварре (C Andrews Lavarre)

О Зависание было вызвано переключением видеовывода или Вам просто казалось, что машина зависла, потому что видеовывод направлялся на несуществующее устройство? Во втором случае, Fn+F8 должно переключить Вас обратно на внутренний дисплей. В *X.org* есть способ работы с двумя мониторами одновременно, он называется MergedFB. Его работа похожа на *Xinerama*, но в MergedFB есть режим 'clone', позволяющий направлять одно и то же изображение на оба монитора. Этот режим подойдет, если Вы захотите вывести изображение на внешний монитор или проектор.

Вам потребуется добавить следующие строки в секцию Device файла **/etc/X11/xorg.conf**:

```
Option "MergedFB" "auto"
Option "CRT2Position" "clone"
Option "MetaModes" "1024x768 800x600 640x480"
```



Благодаря *MergedFB*, Linux должен работать на этом телевизоре.

Первая строка сохраняет возможность переключения конфигураций – если при запуске *X*-сервер обнаружит внешний монитор, то включится режим *MergedFB*. Вторая строка говорит о том, что изображение на внешнем мониторе будет копией изображения на встроенном дисплее. Третья строка –список видеорежимов, они должны быть такими же, какие указаны у Вас в секции Screen. Если видеорежимы не указаны, добавьте только тот видеорежим, в котором Вы работаете и который поддерживается внешним монитором. Чтобы на внешний и встроенный дисплеи выводилось одно и то же, Вам может понадобиться перенастроить BIOS.

В Ваш ноутбук встроена видеокарта на базе чипа ATi, и я точно знаю, что *X.org*-драйвер для нее поддерживает режим *MergedFB*, у меня у самого такой же на iBook. Дополнительную информацию об использовании режима *MergedFB* можно найти на сайте: www.winischhofer.at/linuxsispart2.shtml#mergedfbmode. ***НБ***

Ящик без головы

В Я настроил удаленный рабочий стол (через Центр Управления Xandros) для ныне обезглавленной машины, и теперь не могу вспомнить пароль. Поиск в Google ничего не дал (сайт VNC советует применить *Vncpasswd*, но такой команды на удаленной машине не оказалось). Я не могу подключить к машине монитор, так что Xandros Control Center для смены пароля не запустить. Я думаю, что в Xandros используется обычное удаленное соединение KDE; как можно сбросить пароль из командной строки?

Скут (Scoot), с форума LXF.

О При условии, что на удаленной машине запущен SSH-сервер, для сброса пароля есть несколько способов. Если используется обычное разделение рабочих столов KDE, нужно просто удалить все строки с password из файла **~/.kde/share/config/krfbrc**. Пароль будет сброшен. После этого Вы сможете соединиться с машиной и установить новый пароль через Центр Управления KDE.

Другой вариант – подключиться к этому компьютеру через *ssh* с ключом **-X** и запустить **kcontrol**. Если конфигурация

SSH на «безголовой» машине разрешает X forwarding, запустится Центр Управления KDE, и в нем Вы можете сменить пароль. Если у Вас на рабочей машине установлен KDE, мы рекомендуем сохранять пароли через Менеджер Паролей KDE (*KWalletmanager*). Тогда можно забыть пароли на удаленное подключение и жить спокойно... пока помните пароль от *KWallet*. ***НБ***

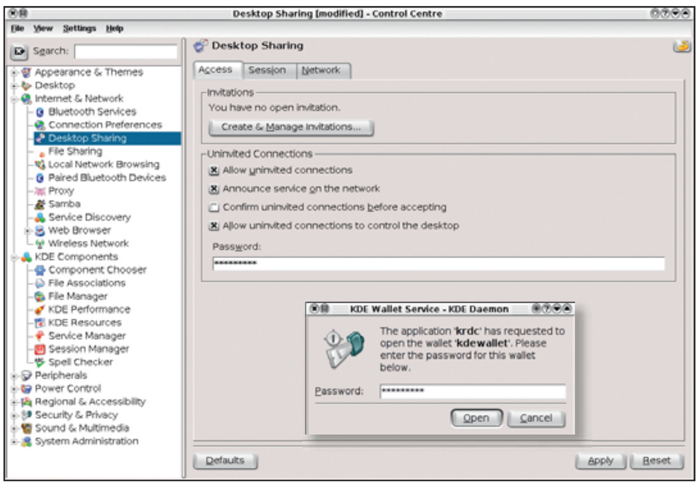
Пакеты для Fedora

Я собрался попробовать Fedora Core 4, на вид чрезвычайно интересную ОС. К сожалению, я не в курсе, какие у нее репозитарии. Я знаю про Fedora Extras и про Livna, но читал в сети, что их еще множество. Некоторые говорят, что пакеты из этих репозитариев нельзя смешивать с Fedora Extras и Livna. Можете ли вы рассказать, какие репозитарии используются чаще всего и содержат больше всего пакетов для архитектуры i386?

Майкл Остин (Michael Austin)

О Про Livna (**http://rpm.livna.org**) Вы уже знаете. Другой заслуживающий внимания репозитарий – FreshRPMs (**http://freshrpms.net**). Большинство репозитариев содержит совместимые пакеты, но правда и то, что между пакетами из различных репозитариев бывают нестыковки, из-за различий в методах сборки пакетов.

Безопаснее всего установить проверенную конфигурацию *Yum*, содержащую только те репозитарии, которые заведомо совместимы. Вы можете ознакомиться с информацией о них в неофициальном FAQ по Fedora (**www.fedorafaq.org/#yumconf**). Следуя инструкциям на этой странице, Вы сможете настроить *Yum* на репозитории Dag, Dries и redhat-kde, в дополнение к упомянутым. Это позволит Вам иметь самый широкий спектр пакетов, избежав всяческих несовместимостей. ***НБ***



Можно запустить Центр Управления KDE через SSH-соединение и сбросить пароль, а KDE Wallet поможет вам не забыть новый.

Спецпакеты

Я администратор сервера на Red Hat Enterprise Linux 4, это хост десяти доменов для сайтов и почты. Для облегчения обновления системы через *Red Hat Network* все программы установлены из RPM-пакетов. Недавно меня попросили перекомпилировать PHP с поддержкой Mcrypt, однако это значит, что мне придется пересобирать PHP при каждом обновлении пакета. Можно ли как-нибудь включить поддержку *Mcrypt* без пересборки пакета, или хотя бы добавить пакеты с php в **pkgSkipList** для *up2date*?

Тони Харрис (Tony Harris)

О Вам повезло! Недавно я наткнулся на проект PHPRPMs (**http://phprpms.sourceforge.net**), предоставляющий PRM-пакеты с PHP, скомпилированным с малоиспользуемыми или не-GPL расширениями. Пакеты в данный момент доступны для Fedora Core 2–5, RHEL 3 и RHEL 4 (i386 и x86_64). Скачав подходящий пакет *php-mcrypt* для RHEL 4, просто установите его командой *rpm*, а после этого перезапустите сервис *httpd*. Если у Вас нет *libmcrypt*, взять RPM-пакет для RHEL можно с сайта **http://dag.wieers.com/packages/libmcrypt**.

Теперь вместо пересборки PHP после обновления с RHN, Вам нужно будет проверять эти два сайта каждый раз, когда захочется обновить *libmcryp* и *php-mcrypt*. ***КК***

Autopackage

Давно хочу установить *Autopackage 1.0* с диска *LXF*, но в журнале не написано, как это сделать. Я скопировал файл к себе в домашнюю директорию, а что дальше? Это что-то вроде tar-файла, с которым нужно как-то поколдовать? Я бы с радостью применял *Autopackage*: он, похоже, воплотил мои мечты о простоте установки ПО. RPM-пакеты я умею устанавливать, но мне

>>

КРАТКАЯ СПРАВКА: ФАЙЛОВЫЕ СИСТЕМЫ

Файловая система – это метод организации хранения информации на запоминающих устройствах, например, жестком диске или DVD. Все файловые системы делают по сути одно и то же, но разными способами, и каждая из них имеет свои слабые и сильные стороны.

Много лет в Linux была доступна одна файловая система – ext2, довольно надежная, быстрая и проверенная временем. Одна из причин быстродействия – отсутствие журналирования. Говоря по-простому, журнал – это протокол последних изменений в файловой системе, который делает намного более простым ее восстановление в случае некорректного завершения работы. Файловая система ext2 после сбоя довольно долго проверяется программой *fsck*.

Добавление журналирования в ext2 дало нам файловую систему ext3. Две другие популярные файловые системы, предоставляемые как альтернатива во многих дистрибутивах – ReiserFS и XFS. Первая использует принципиально иной метод доступа к файлам, более эффективный при хранении множества файлов небольшого размера.

XFS использует кэширование больше других систем, поэтому она очень быстрая, но при выключении питания может вызвать большую потерю данных. Это хороший выбор для машин, подключенных к блокам бесперебойного питания (UPS), или для ноутбуков. Есть и другие файловые системы, вроде JFS от IBM и Reiser4, но они не очень широко используются (пока).

Файловые системы Windows тоже заслуживают упоминания. Даже если у вас нет Windows, вы можете использовать устройство (например, MP3-плейер), с файловой системой FAT из Windows, которая читается и пишется в Linux без проблем. В файловую систему NTFS, используемую по умолчанию Windows XP, Linux не может производить безопасную запись. Многие дистрибутивы при установке распознают NTFS-разделы, но подключают их только для чтения.

ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ: БРАНДМАУЭРЫ

FAQ Что такое брандмауэр, и за каким пожаром он может мне понадобиться?

Брандмауэр – это комбинация из аппаратных и программных средств, защищающая компьютер(ы) от несанкционированного доступа. Самое распространенное применение – защита компьютера или сети от сетевых вандалов. От пожара в офисе брандмауэр вас не спасет.

FAQ И аппаратных, и программных? Так это программа или железяка?

Бывает и так, и этак. Аппаратные брандмауэры обеспечивают лучшую защиту, поскольку они останавливают злоумышленника еще до того, как он достучался до компьютера. Множество роутеров снабжено встроенными брандмауэрами. Старый компьютер можно превратить в брандмауэр, установив на него специальный дистрибутив вроде IPCop. Его потянет даже 486-й процессор (брандмауэр не требует особой производительности).

FAQ У меня только один компьютер. Могу я на нем запустить брандмауэр?

Да, можете, и очень эффективный. Конечно, не такой, как аппаратный, но налетчиков отгонит. Истинные параноики используют оба.

FAQ Какие есть варианты?

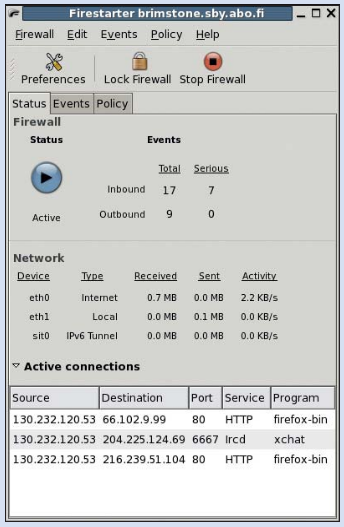
Есть брандмауэр, встроенный в ядро Linux – *iptables*. Его можно настроить, создав правила, какие пакеты пропускать, а какие нет. Написание правил – штука сложная, и у вас может получиться вовсе не то, что вы хотели. Есть много программ, облегчающих настройку брандмауэра, например, *Shorewall* (**www.shorewall.net**), *Firestarter* (**www.fs-security.com**), *Guarddog* (**www.simonzone.com/software/guarddog**) и *Firewall Builder* (**www.fwbuilder.org**). Все эти программы создают правила для *iptables*, и вы можете использовать их для генерации скрипта, который потом можно будет применить на сервере без *X*.

FAQ А разве Linux не безопасен?

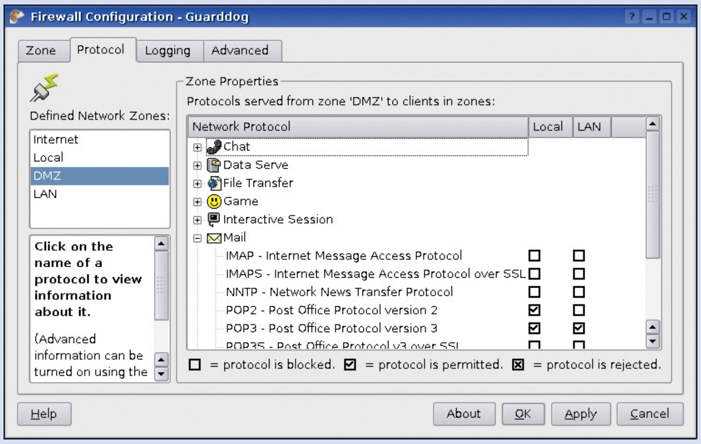
Да, но только если вы сами об этом позаботитесь. Замки на ваших дверях не имеют смысла, если вы их не запираете. Linux предоставляет все средства для защиты, а пользоваться ими должны вы. Также не забывайте про права доступа.

FAQ У меня и украсть нечего, разве не так?

Украсть можно больше, чем вы думаете. Кэш вашего браузера может рассказать о посещаемых вами финансовых сайтах – паролей там нет, но информация пригодится вору. В вашем почтовом ящике – конфиденциальная переписка. Пароли от вашего Интернет-провайдера… Кроме того, взломанную машину могут использовать для рассылки спама. Причем все это – без всяких прав суперпользователя.



Firestarter (справа) и *Guarddog* (внизу) – две из множества программ Linux для составления правил брандмауэра.



<<

ни разу не удавалось заставить работать программу из tar-файла. Мой дистрибутив – Mandrake 10.1 PowerPack.

Дж. А. Дрейк (J A Drake)

О *Autopackage* придуман как раз для упрощения пользования, его даже не надо устанавливать. Когда Вы попытаетесь что-нибудь установить из .package-файла, он загрузит последние версии всех необходимых файлов. Вам не нужно копировать Autopackage на жесткий диск, и это явно указано в документации, которая лежит в директории **Autopackage** нашего диска. Чтобы посмотреть его в работе, запустите терминал, перейдите в режим суперпользователя, набрав **su -**, и выполните

```
bash /mnt/cdrom/Magazine/HotPicks/Autopackage/autopackage-qt-1.0.x86.package
```

Вас спросят, не хотите ли Вы скачать *Autopackage*? Ответьте 'Y'. После этого все необходимые файлы будут загружены и установлены.

Установка из tar-архивов на первый раз потребует терпения, но потом все будет проще. Для начала распакуйте архив, одной из следующих команд (в зависимости от типа архива – gzip или bzip2):

```
tar xzf somepackage.tar.gz
tar xjf somepackage.tar.bz2
```

В распакованной директории Вы скорее всего найдете файлы **INSTALL** и **README**, содержащие инструкции по установке.

В PRM-дистрибутивах вроде Mandrake установка стопорится обычно на стадии **./configure**: выскакивают сообщения об отсутствии библиотек, хотя менеджер пакетов ясно показывает, что на самом деле они есть. Это потому, что RPM-пакеты разделены на два вида: стандартный RPM с программой или библиотекой и '-devel' пакет, содержащий заголовочные файлы.

Для работы программ они не нужны, но нужны для компиляции программ из исходных текстов. Так что если *configure* выдаст сообщение об отсутствии *libfoo*, проверьте, установлен ли пакет *libfoo-devel*. ***НБ***

Сюда – Sudo

В Я пытался создать скрипт для автоматизации некоторых операций, но не смог сделать так, чтобы часть его выполнялась от лица суперпользователя, а все остальное – от лица того, кто его запустил. Я не хочу запускать весь скрипт из-под root, только отдельную его часть. Думал воспользоваться su, но потом сообразил, что после этого я буду уже не я и собственный скрипт выполнять не смогу.

Я понимаю, что su – не лучший выход, но для тестирования сошло бы; или использование su из скрипта – вообще полная чушь? Другой вариант – запустить еще одну копию оболочки под root, но я не уверен, что это можно сделать из скрипта.

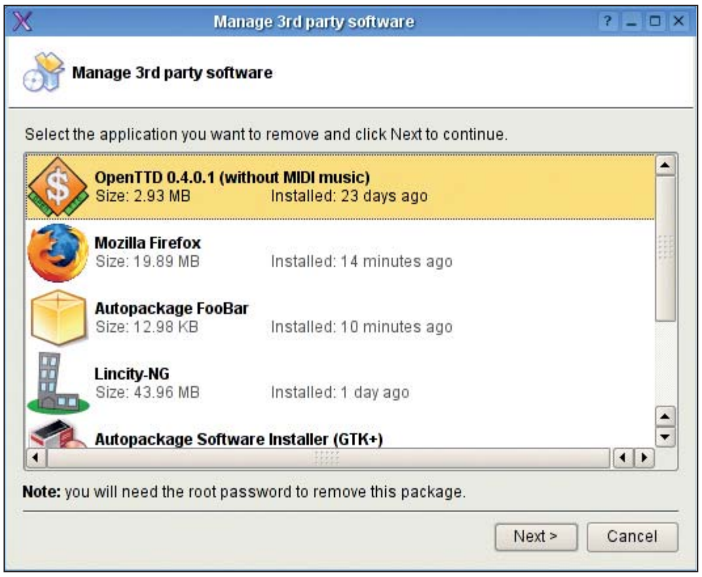
MGE, с форума LXF

О Команда su создает новый процесс с правами другого пользователя, а сам скрипт останавливается и ждет завершения этого процесса. Использование su из скрипта – плохая идея, угрожающая безопасности, поэтому оно часто блокируется. Лучше взять *sudo*. Эта команда разрешает некоторым пользователям выполнять определенные программы с правами root, не зная пароля. По умолчанию sudo запрашивает пароль запускающего ее пользователя, но для некоторых команд это можно отключить.

Укажите полный путь к нужным программам в файле **/etc/sudoers** и добавьте опцию 'NOPASSWD', если не хотите, чтобы скрипт останавливался и спрашивал пароль. Вот запись, позволяющая пользователю fred монтировать и размонтировать файловые системы без пароля:

```
fred ALL = NOPASSWD: /bin/mount,/bin/umount
```

Обратите внимание на комментарий в самом начале файла **/etc/sudoers** – он должен редактироваться только командой **visudo**. Запустите *visudo* из-под root, и стартует редактор, указанный в перемен-



Autopackage **запросто устанавливает программы – и даже самого себя.**

ной окружения **$EDITOR**. Можно отвести редактор специально для visudo:

```
EDITOR=kate visudo
```

Это делается неспроста. *visudo* не работает с файлом **/etc/sudoers** напрямую: сначала делает копию, проверяет ее на ошибки, а уж потом заменяет на нее **/etc/sudoers**. Большинство экспертов считают такую тактику хорошей, поскольку случайные опечатки могут причинить вред Вашей системе. ***НБ***

Журнальный вариант

В У меня есть несколько скриптов, и я хочу, чтобы они генерировали файлы журналов отладки, причем я мог бы включать и выключать журнал по желанию. Что бы вы посоветовали?

Трент Хорнибрук (Trent Hornibrook)

О А Вы не пробовали *Syslog*? И PHP, и C, и Perl имеют необходимые библиотеки для отсылки сообщений *Syslog* на хост. Оттуда Вы можете настроить *Syslog* на запись определенных сообщений в отдельный файл, а *Logrotate* – на обновление журналов.

Сперва добавьте новый селектор (левая сторона) и лог-файл (правая сторона) в **/etc/syslog.conf** и перезапустите *Syslog*. Селектор состоит из уровня разрешения и запретов. Уровень разрешения может быть local1 – local7 из пока не определенных. В данном случае я выбрал local3 и разрешил запоминание всех сообщений:

```
# newprog
local3.* /var/log/newprog.log
root$ service syslog restart
```

Если нужно, чтобы в журнал записывались только ошибки (**err**), **syslog.conf** должен содержать что-то наподобие

```
# newprog
local3.err /var/log/newprog.log
```

Теперь убедитесь, что *Syslog* работает нормально – запустите следующее и проверьте вывод в журнал:

```
$ logger -p local4.err test message
$ tail -f /var/log/newprog.log
Jan 31 1 1:46:40 host user: test message
```

Убедившись, что с *Syslog* все в порядке, можно настроить *Logrotate* для ротации файлов журналов, задав ему соответствующие правила; внесите полный путь к новому лог-файлу в конфигурационный файл *Logrotate*. Например, под Red Hat Enterprise Linux нужно поправить **/etc/logrotate.d/syslog**, добавив **`/var/log/newprog.log'** в первую строку файла конфигурации.

Теперь все, что Вам нужно – вызвать *Syslog* из Вашего скрипта, используя предварительно созданный селектор, который Вы добавили в **syslog.conf**. На других уровнях Вы можете выключить частичное слежение, добавив уровень запрета, как описано выше.

Более подробную информацию о работе с *Syslog* из PHP, Perl, C и *Bash*, Вы можете получить из следующих источников man-страниц:

Sys::Syslog (3pm)
Unix::Syslog (3pm)
logger (1)
syslog (2)

Также рекомендую заглянуть на сайт http://php.net/syslog. ***КК***

Blue Skype

В Я приобрел новый мобильный телефон с поддержкой Bluetooth. К нему еще прилагается Bluetooth-гарнитура. Можно ли использовать ее с компьютером? Я был бы счастлив, если бы по ней можно было бы разговаривать через *Skype* или другой Интернет-телефон. И если можно, то как?

С форума LXF

О Во-первых, Вам понадобится Bluetooth-адаптер, если только он не встроен в Ваш компьютер. В любом компьютерном магазине он стоит чуть больше 10 долларов. Нужно будет установить пакет **bluez**, содержащий драйверы и утилиты для работы с Bluetooth – они есть в большинстве дистрибутивов. Если у Вас установлена рабочая среда KDE, запустите **kbluetoothd**, и в системном меню отобразится иконка доступных Bluetooth-подключений. Щелчок по этой иконке даст Вам доступ к Bluetooth-устройствам.

Для синхронизации контактов мобильного телефона есть несколько программ. Если Ваш телефон – Nokia, вам подойдет *Gnokii*, тоже имеется практически в любом дистрибутиве или на www.gnokii.org. Более независимый от марки телефона подход поддерживается *KMobileTools* (http://kmobiletools.berlios.de).

Пакеты, прилагающиеся к дистрибутиву, давно устарели. Чтобы получить новые функции, например, резервное копирование и восстановление телефонной книги, Вам придется собрать версию из SVN. Не волнуйтесь, это очень простая процедура, отлично описанная в имеющемся на сайте HOWTO.

Использовать гарнитуру для *Skype* тоже можно, но скорее всего в Вашем дистрибутиве не окажется нужных пакетов. Проект *Bluetooth-alsa* (http://bluetooth-alsa.sourceforge.net) предоставляет поддержку Bluetooth-гарнитур как ALSA-устройств, при этом система считает гарнитуру звуковой картой. Затем Вам надо будет настроить *Skype* или другую программу на использование этой "звуковой карты". Вы можете даже слушать MP3 через вашу гарнитуру – правда, не ждите выдающегося качества звука.

Скачайте и установите ПО, как описано на сайте. Затем установите гарнитуру в парный режим и подключите ее следующими командами:

```
modprobe -v snd_bt_sco
esdctl stop
hcitool scan
btsco [адрес]
```

[адрес] в последней команде нужно взять из вывода команды **hcitool scan**. Это будет что-то вроде 00:13:EF:00:09:44. Теперь можно протестировать гарнитуру:

```
ls -l /proc/asound
aplay -D plughw:Headset somesound.wav
```

Первая команда должна показать все записи для гарнитуры, вторая проиграет на нее звуковой файл **somesound.wav**. Если все заработало, большинство этих действий можно автоматизировать. Например, добавить имя модуля в список автоматически загружаемых (**/etc/modprobe.conf** или **/etc/modules.conf**). Найдите строку, начинающуюся с 'alias snd-card-0' и добавьте после нее:

```
alias snd-card-1 snd_bt_sco
```

Команды *esdctl* и *btsco* можно добавить в короткий скрипт, вызываемый каждый раз при подключении гарнитуры:

```
#!/bin/sh
esdctl stop
btsco 00:13:EF:00:09:44
```

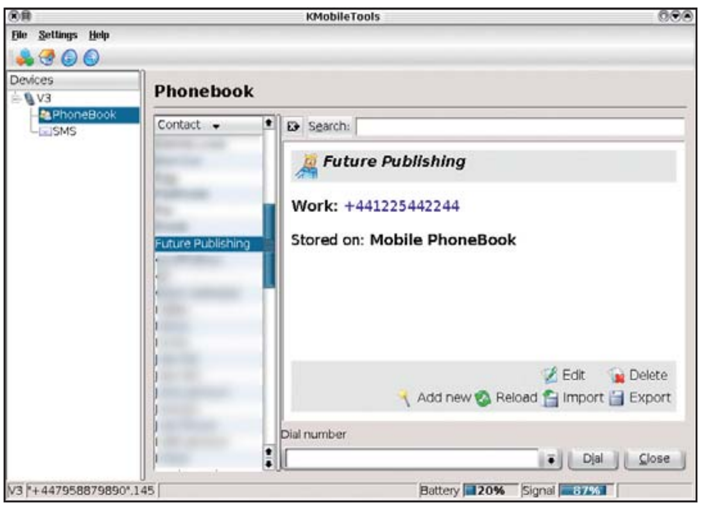
Только подставьте верный адрес Вашего устройства. ***НБ***

Кстати о рыбках

В Знаю, что это глупо, но я хотел бы установить *Xfishtank* в качестве фона для рабочего стола. У меня установлена Mandriva 2006 Free с KDE и *Xfishtank* из RPM-пакета. Но все, чего я достиг – мимолетное появление рыбок при выключении компьютера. А я бы хотел видеть их постоянно. Как это сделать?

Джобекс (Jobex), с форума LXF

>>



Пользуйтесь ***KMobileTools*** **для просмотра, синхронизации и восстановления телефонной книги вашего мобильника. И читать SMS из нее тоже можно.**

<<



Почувствуйте маленькие радости организмов с крошечным мозгом: просто плавают туда-сюда, туда-сюда, пуская пузыри... *Xfishtank*.

О В KDE есть собственное корневое окно, запускающееся поверх корневого окна X-сервера, и программы, отображающие что-то на окне *X*-сервера, им заслоняются. Вы можете увидеть их лишь за небольшой промежуток времени, когда KDE уже завершил работу, а *X*-сервер еще нет.

К счастью, решение очень простое. Щелкните правой кнопкой мыши на рабочем столе и выберите пункт "Настроить рабочий стол" (Configure Desktop). Зайдите в пункт "Поведение" (Behaviour) и включите "Позволить программам выполняться в родительском окне" (Allow Programs In Desktop Window) – учтите, что при этом нельзя менять назначение правой кнопки мыши по умолчанию. То же самое можно сделать, зайдя в Центр Управления KDE. ***НБ***

Нас атакуют!

В Я отвечаю за web- и FTP-сервер под управлением Red Hat Enterprise Linux 4. Недавно я настроил *Logwatch* на посылку отчетов и нашел записи, которые меня беспокоят. Они касаются неудачных попыток авторизации и несуществующих пользователей.

Такие записи появляются что ни день, но IP-адреса и число попыток каждый раз разные. Похоже, кто-то пытается подобрать пароль к моей системе. Как я могу пресечь эти попытки? Какие действия вы порекомендуете для улучшения безопасности сервера?

Д.З. Флинт (DZ Flint)

О Вы правы! Записи, которые послал вам *Logwatch* – неудавшиеся попытки проникновения. Есть несколько способов ужесточить доступ к *sshd*-серверу. Откройте конфигурационный файл *OpenSSH* (**/etc/ssh/sshd_config**). Давайте взглянем на него.

Параметр AllowUsers позволит входить в систему только оговоренным пользователям. Для FTP-сервера с множеством пользователей это вариант достаточно сложный, и все-таки я бы рекомендовал именно его, хотя атакующие все равно будут пытаться угадать пароли любого пользователя. Также, запретите root-доступ по ssh, используя следующую опцию:

```
PermitRootLogin no
```

Используйте сильные пароли. Впрочем, чтобы их не угадали, я бы посоветовал перейти на аутентификацию по ключам, генерируемым утилитой *ssh-keygen*. Разберитесь с ключами при помощи *ssh-agentissh-add* и откажитесь от системы паролей.

Число атак на *sshd*-сервер можно уменьшить, назначив серверу другой порт по умолчанию. Большинство атакующих проверяет лишь стандартный порт 22. Попробуйте изменить его, например, на 222. Разрешайте пользоваться только версией 2 протокола SSH, в первой версии есть уязвимости. Убедитесь, что запрещены пустые пароли: предусмотрите опцию `PermitEmptyPasswords no'.

После запоминания всех этих изменений *sshd*-сервер нужно перезапустить.

Эти действия избавят от абсолютного большинства атак. Однако статичных правил недостаточно, ибо атаки изменчивы по природе. Кроме того, шифрование с публичным ключом может Вам не подойти. Проблемой занимается множество другого открытого ПО (*Sshdfilter*, *Blockhosts* и пр.), привлекая для работы *PortSentry*, *Iptables*, *Tcpwrappers* и др. Я сфокусируюсь на DenyHosts (**http://denyhosts.sourceforge.net**), так как он работает с *Tcpwrappers*, который есть в большинстве Unix-систем.

Для использования с *Tcpwrappers*, *sshd* должен быть скомпилирован с поддержкой *libwrap*. Собрана ли ваша версия *sshd* с *libwrap*, Вы узнаете, выполнив команду

```
# ldd /usr/sbin/sshd lgrep libwrap
libwrap.so.0 => /usr/lib/
libwrap.so.0 (0x00140000)
```

DenyHosts написана на Python – значит, потребуется интерпретатор этого языка. Вы можете установить его следующим образом:

```
# up2date -i python
```

Теперь скачайте самую свежую версию. Для RHEL 4 доступен RPM-пакет:

```
rpm -ivh http://kent.dl.sourceforge.net/
sourceforge/denyhosts/DenyHosts-1.1.3-
python2.3.noarch.rpm
```

DenyHosts поставляется с простым конфигурационным файлом (**/usr/share/denyhosts/denyhosts.cfg-dist**), его можно использовать как шаблон для Вашей системы. Вот некоторые опции, которые Вы, возможно, захотите изменить:

- **PURGE_DENY = 1w**

Запрещает удалять записи, которым меньше одной недели (1w).

- **DENY_THRESHOLD_INVALID = 5**

После скольких попыток начинать блокировку для нелегальных пользователей

- **DENY_THRESHOLD_VALID = 10**

После скольких попыток начинать блокировку для легальных пользователей

Если Вы хотите получать уведомления о новых блокируемых хостах, введите свой адрес в строке `ADMIN_EMAIL = webmaster@example.com'.

Конфигурационный файл очень хорошо прокомментирован и несложен для восприятия. Когда Вы определитесь со всеми опциями, скопируйте файл в **/etc/denyhosts.cfg**.

Теперь нужно подправить инициализационный скрипт (**/usr/share/denyhosts/daemon-control-dist**) для нашей системы:

```
DENYHOSTS_CFG = "/etc/denyhosts.cfg"
```

и установить этот скрипт:

```
# cp /usr/share/denyhosts/daemon-control-
dist /etc/init.d/denyhost
# chmod +x /etc/init.d/denyhost
# chkconfig denyhost on
```

Теперь можно запустить *DenyHosts* следующей командой:

```
# /etc/init.d/denyhost start &
```

Отныне, если Вы увидите в логах следы попыток проникновения в систему, знайте, что хост злоумышленника уже заблокирован!

```
#tail -n3 /etc/hosts.deny
sshd: 66.34.205.1
sshd: 64.34.193.58
sshd: 220.1 17.241.3
sshd: 218.85.1 19.83
```

Письмо с уведомлением будет отправлено на адрес, указанный в ADMIN_EMAIL.

Безопасность сервера – довольно обширная тема. Попробуйте поискать в сети по ключевым словам "defending against brute force ssh attacks". ***КК***

Черный экран Debian

В Сегодня я установил Debian 3.1 с диска *LXF 70*. Малость помучившись, я установил первый диск; все, что мне теперь надо – открыть систему и продолжить.

При входе я ввожу свое имя пользователя, пароль... и попадаю в черный экран с надписью `david@debian~$:' . Я что-то сделал неправильно или нужно произнести какое-то заклинание? Я новичок, но надеюсь, что установил систему не зря.

Дэвид Чарлтон (David Charlton)

О Вы установили только базовую систему, которая не включает графическое окружение. На втором этапе установки, после перезагрузки, вас спрашивали про наборы пакетов. Первым набором был Рабочий стол, кажется, что он отмечен по умолчанию, потому что подсвечен – а вот и нет, устанавливаются только те наборы пакетов, слева от которых стоит звездочка. Если же Вы просто нажали Enter, то ничего не установилось.

Но это не повод паниковать или бросаться переустанавливать систему. Войдите в систему как суперпользователь и введите команду **aptitude**; загрузится менеджер пакетов. Перейдите на пункт задач (Tasks), нажмите Enter. Спуститесь на пункт End-User и снова нажмите Enter. Теперь выберите Desktop Environment и нажмите +. А теперь нажмите G (увидите, что будет установлено) и еще раз G для подтверждения установки. Эти действия приведут к установке KDE и Gnome, и при первом входе в систему Вы сможете выбрать одно из этих сред.

Во время установки вам зададут несколько вопросов о настройках (если Вы не уверены в своих знаниях, согласитесь на ответы по умолчанию, они прекрасно работают). Те же вопросы были бы заданы при инсталляции системы, если выбрать набор пакетов Рабочий стол. Графическое окружение запустится при следующей загрузке системы. ***НБ***

Несговорчивый QEMU

В Здравствуйте, уважаемая редакция! Я пыталась установить QEMU 0.7 и 0.8 с дисков журнала на Mandriva Linux 2006. При это компьютер стабильно выдает мне следующую ошибку:

```
> $ qemu -boot d -cdrom /dev/cdrom
> Could not get DNS address
> $
```

Я пробовала менять параметры командной строки – ничего не вышло Помогите, пожалуйста!

Аня

★ ВОПРОС МЕСЯЦА!

Редакция вместе с компанией Bitrix выбрали самый интересный вопрос месяца и поздравляют победителя, ***Никиту Неверова***, с заслуженной наградой – **сертификатом на 1000 рублей** в LinuxCenter.ru

APT через посредника

Приветствую вас, редакция Linux Format. Вопрос в следующем: в номере LXF 2(76) было много сказано об утилите управления пакетами apt, с помощью которой можно производить обновление базы данных пакетов, обновление самих пакетов, установку новых программ с разрешением зависимостей. Эта прекрасная утилита является одной из сильных сторон дистрибутивов на базе Debian, но вот незадача: допустим у меня есть хороший Интернет-канал, работающий через proxy-сервер с аутентификацией по паролю! В этом случае возникают вопросы по настройке apt через proxy, да ещё и с аутентификацией! Подскажите что делать в этом случае для дистрибутивов Ubuntu и Debian, и есть ли различия в настройке этих параметров у данных дистрибутивов. Заранее спасибо.
Никита Неверов

Напрашивается примерно такой универсальный вариант решения – определить переменную среды:

```
HTTP_PROXY="http://login:password@xxx.xxx.xx.xx:yyyy"
export HTTP_PROXY
```

где xxx.xxx.xx.xx – адрес proxy-сервера, yyyy – порт, а login и password – сооветствующие параметры Вашей учетной записи. В Debian такую строку резонно вписать в профильный файл суперпользователя, а в Ubuntu и Kubuntu, в которых учетной записи root как таковой по умолчанию нет, эту переменную можно определить в профильном файле обычного пользователя (в **~/.bashrc**). Ну и, конечно, никто не запрещает сделать это просто из командной строки:

```
export HTTP_PROXY="http://login:password@xxx.xxx.xx.xx:yyyy"
```

Есть и другой способ, более отвечающий Debian Way – отредактировать файл **/etc/apt/apt.conf**. По умолчанию он содержит строку

```
Acquire::::Proxy "false";
```

предписывающую не использовать proxy. Ее достаточно изменить следующим образом:

```
Acquire::http::Proxy "http://login:password@xxx.xxx.xx.xx:yyyy"
```

К слову сказать, указание на порт в большинстве случаев можно и опустить. ***АФ***

ВОПРОС МЕСЯЦА! ★

QEMU Выдает ошибку потому, что не может найти адрес DNS-сервера, который превращает строковые имена сайтов (www.linuxformat.ru) в IP-адреса (83.222.7.132) и нужен QEMU для внутренних нужд. Обычно адрес DNS-сервера выдается автоматически при подключении к Интернету или же назначается провайдером и Вы прописываете его в системе вручную. Если Ваш компьютер не подключен к Интернету, попробуйте добавить в файл **/etc/resolv.conf** следующую строку (это нужно делать от имени root):

```
nameserver 127.0.0.1
```

Это заставит QEMU замолчать, по крайней мере, до тех пор, пока ему не потребуется реально воспользоваться DNS. ***ВС***

GPRS без проблем

Позвольте сразу второй вопрос: Я использую для выхода в Интернет сотовый телефон "Samsung SGH X450" (GPRS), это единственная причина из-за которой я до сих пор пользуюсь Windows! Я люблю Linux, быстро привыкла к нему, но не могу выйти в Интернет через сотовый из Mandriva Linux 2006. Пожалуйста, подскажите, как мне заставить Mandriva Linux 2006 использовать сотовый как модем, и тогда я смогу полностью отказаться от Windows 2000!
Аня

А какой у вас кабель? Если это COM, то все очень простО после подключения телефона он будет виден в системе под именем **/dev/ttyS[номер COM-порта]**. Скорее всего это будет **/dev/ttyS2**. Имя этого устройства и нужно указать в KPPP. Для некоторых телефонов обязательно нужна строка инициализации модема **(AT+CGDCONT=1,"IP","<acm вашего оператора>")**. Все эти параметры знать не обязательно, если установить GPRS EasyConnect (http://gprsec.hu/modules/index/): тут нужно просто выбрать из списка телефон, порт для подключения (есть функция автоматического обнаружения порта телефона) и вашего оператора. Samsung SGH X450 находится в списке поддерживаемых аппаратов (через COM-кабель).

Если же у вас USB-кабель, то проблем будет немного больше – для разных типов кабелей нужно указывать разные устройства. После подключения проверьте, не появился ли у вас в системе файл **/dev/ttyACM0**. Если да, то это и есть ваш телефон. Но, скорее всего вы приобрели USB-конвертер, и телефон будет известен системе как **/dev/ttyUSB0**. Если этого устройства нет, попробуйте вручную загрузить модуль pl2303 (modprobe pl2303). ***АК*** LXF

СПРАШИВАЙТЕ ПРАВИЛЬНО!

- Пожалуйста, не забывайте сообщать все необходимые данные о вашей системе. «У меня не работает X» нам мало что скажет, если мы не знаем, какую версию X вы имеете в виду и на какой аппаратной конфигурации пытаетесь ее запустить.
- Опишите свою проблему с максимальной точностью. Причитания типа «Оно не работает» или «У меня ошибка» вряд ли дадут нам возможность помочь Вам. Каким именно образом нечто не работает? Чего вы от него хотели? Какое поступает сообщение об ошибке?
- Помните, пожалуйста, что люди, работающие в нашем журнале, НЕ являются авторами или разработчиками Linux или Вашего конкретного пакета (дистрибутива). Иногда нужная Вам информация имеется на соответствующем сайте. Ознакомьтесь с документацией!

Мы стараемся ответить на все вопросы. Если вы не нашли ответа на свой, просмотрите другие ответы – может быть, там разбирается проблема, аналогичная вашей. Все вопросы, к сожалению, поместить невозможно.

Диск Linux Format

В этом месяце на нашем диске не одна, не две, а целых три операционных системы!

FOOBUNTU?



Майк Сондерс только что на спор установил Coherent на ноутбук с 486-м процессором.

Сейчас Ubuntu – один из моих любимейших дистрибутивов. Солидный фундамент Debian, стилизация графического интерфейса Gnome и безвариантный выбор приложений для установки – разработчики действительно фокусируются на освежающей простоте. (Допустим, я пользуюсь не Gnome, а *IceWM*, но все равно рекомендую этот дистрибутив как полноценную ОС рабочего стола.) Однако меня беспокоит, что растущая волна дистрибутивов-производных от Ubuntu снижает ценность брэнда.

«СЕЙЧАС НАМ В МИРЕ LINUX НУЖНА НЕКАЯ КОНСОЛИДАЦИЯ.»

Kubuntu, Edubuntu, Xubuntu... да, они используют общий репозитарий, но это разные операционные системы, и, что самое главное, нацелены на разные платформы.

Что нам сейчас нужно в мире Linux, так это некоторая консолидация дистрибутивов. Широкий выбор – это хорошо, но желательно соблюдать некоторое единообразие, чтобы не сбить с толку начинающих пользователей всеми этими Foobuntu, которые, по сути, представляют собой одну операционную систему. Рядовой пользователь, только что установивший Kubuntu, будет весьма озадачен, получив пакет Ubuntu, которому подавай всю инфраструктуру Gnome.

Безусловно, эта проблема не нова. Все дистрибутивы, предлагающие выбор рабочих столов, в определенной степени с ней сталкиваются. Однако несколько вариантов Ubuntu с разными рабочими столами усугубляют сложность, и я думаю, что для Ubuntu было бы лучше быть единой ОСью, имеющей один фокус и направляющей усилия разработчиков на одну платформу. Мы бы не хотели получить еще и Fluxbuntu, TWMbuntu и Ncursesbuntu!

mike@saunders@futurenet.co.uk

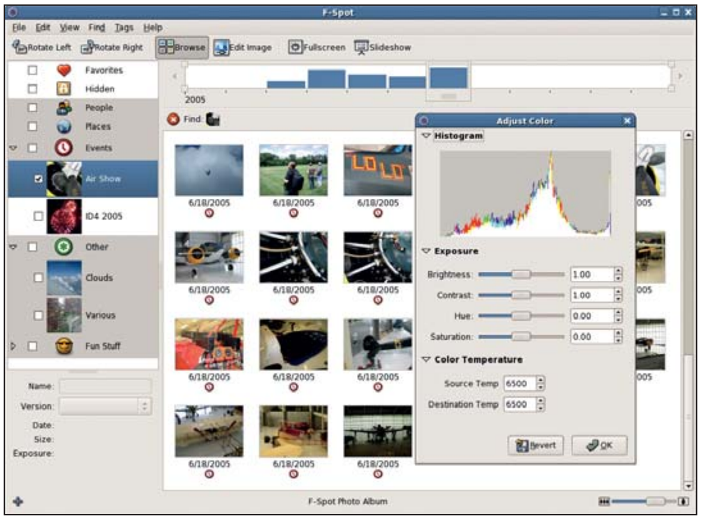
РАБОЧИЙ СТОЛ

MONO-мания!

Mono, .NET-платформа с открытым кодом, продолжает развиваться ускоренным темпом. Когда проект только начинался, многие обозреватели настороженно относились к попытке разработать платформу на технологиях Microsoft, но результаты того стоили. Теперь у нас есть такие отличные приложения, как *Beagle* и *F-Spot*, новая среда разработки и базис, способствующий разработчикам в портировании приложений с Windows на Linux.

Даже если вы игнорируете продукты Microsoft и на совместимость с Windows вам наплевать, Mono в любом случае стоит попробовать. Например, C# – весьма успешный язык, достойный изучения: он совместил элементы C, C++ и Java и поборол некоторую их идиосинкразию. *MonoDevelop* (ищите его в разделе **Разработка** на нашем диске) представляет собой элегантную и богатую интегрированную среду разработки для программирования на C# (и других языках).

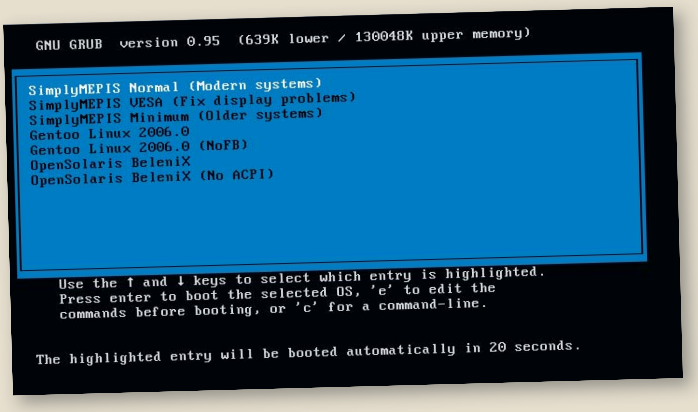
Сам по себе *MonoDevelop* написан на Mono с использованием привязок *GTK#*, чем демонстрирует неплохие способности среды и библиотек. На диске также находится множество других Mono-приложений, раскрывающих возможности платформы. *F-Spot*, утилита управления фотографиями для Gnome, приобрела многочисленных фанатов за свой шикарный интерфейс и ориентированный на пользователя дизайн. По части аудио столь же впечатляет *Banshee*, который умеет проигрывать музыку, записывать компакт-диски, общаться с iPod и отображать альбом.



Главное назначение F-Spot – организация хранения изображений, но он еще и снабжен полезнейшими инструментами обработки фото.

А есть ещё *Blam* – отличная «читалка» RSS; *Muine* – музыкальный проигрыватель, напоминающий *Banshee*; и *Tomboy* – записная книжка для рабочего стола, использующая идеи wiki. Библиотеки Mono и среду времени выполнения (необходимую для запуска приложений Mono) вы найдете в разделе **Система** нашего диска. Запустите файл **mono-1.1.13.4_0-installer.bin** и следуйте инструкциям (более детальная инструкция также находится на диске). Наша коллекция содержит всё, что нужно для запуска Mono, программирования на C# и изучения отличного программного обеспечения Mono.

ЭКСКЛЮЗИВНЫЙ ЗАГРУЗОЧНЫЙ DVD С ТРЕМЯ СИСТЕМАМИ



***Grub* и немного хакинга позволили создать единое меню для загрузки SimplyMepis, Belenix и Gentoo.**

Вполне возможно, что это первый в мире загрузочный DVD, способный загружать сразу три операционные системы. Думаем вы будете очень рады узнать, что для того, чтобы попробовать эти операционные системы, вам не потребуется возиться с записью компакт-дисков. И SimplyMepis, и OpenSolaris, и Belenix могут быть загружены прямо с DVD с помощью меню загрузчика Grub. Просто вставьте диск, перезагрузите компьютер и с помощью курсора выберите интересующую вас систему.

Мало того, мы умудрились втиснуть туда еще и последний релиз одного из основных дистрибутивов: Gentoo 2006.0! Он также является загрузочным; кроме того, все три системы работают как Live CD, то есть запускаются прямо с DVD. Вы можете немедля их попробовать безо всякой установки, а захотите – установите их на свой жёсткий диск. На страницах 46 и 56 можно прочесть о том, как запускать и устанавливать эти операционные системы. Наслаждайтесь!

ДИСТРИБУТИВЫ

НА ДИСКЕ

GENTOO LINUX 2006.0

Gentoo еще никогда не выглядел так круто. Хотя дистрибутив ориентирован на опытных пользователей, не робеющих перед командной строкой, команда Gentoo позаботилась о графическом инсталляторе (мы расскажем о нём немного позже). Для тех, кто не в курсе: идеология Gentoo – скорость и полный контроль, причем предпочтение отдано установке пакетов компиляцией из исходных кодов и массированному конфигурированию множества параметров. Gentoo вполне заслужил славу дистрибутива переднего края – он для тех, кто хочет иметь самые свежие версии установленного ПО и компилировать приложения с личными настройками и оптимизациями.

Несмотря на высокий уровень контроля, разобраться в Gentoo не так уж сложно, благодаря изобилию и дотошности его документации. Даже новичок в Linux сможет выполнить инсталляцию, всего лишь точно следуя инструкциям. В предыдущих версиях Gentoo установка выполнялась из командной строки – пользователи набирали команды или копировали их из инструкции; но скоро эта процедура канет в прошлое, благодаря новому графическому инсталлятору. Релиз Gentoo, записанный на наш DVD, запускается как обычный Live DVD и позволяет испробовать Gnome 2.12.2 и подборку самого свежего ПО без необходимости его установки.

В Gentoo 2006.0 также вошли ядро 2.6.15, *GCC 3.4.4* и улучшенная поддержка для PPC/PPC64. Чтобы установить его на жёсткий диск, вам потребуется выход в Интернет, немного свободного времени и довольно много терпения: хотя документация безмерно полезна, установка Gentoo – не для малодушных, так что если вы только делаете первые шаги в мире Linux, лучше попробуйте SimplyMepis (тоже есть на DVD). Новый графический инсталлятор отнюдь не упрощает установку Gentoo, он задуман так, чтобы установочный процесс проходил немного быстрее и не требовал суетливого копирования команд из инструкции в командную строку.

Разработчики рекомендуют держать под рукой руководство по установке, на нашем DVD вы найдёте его в каталоге **Distros/Gentoo**. Тем, кто никогда раньше не ставил Gentoo, стоит распечатать это руководство (или вывести его на экран соседнего компьютера), поскольку оно прямо-таки досконально. Приводим краткую инструкцию, как приступить к установке Gentoo…

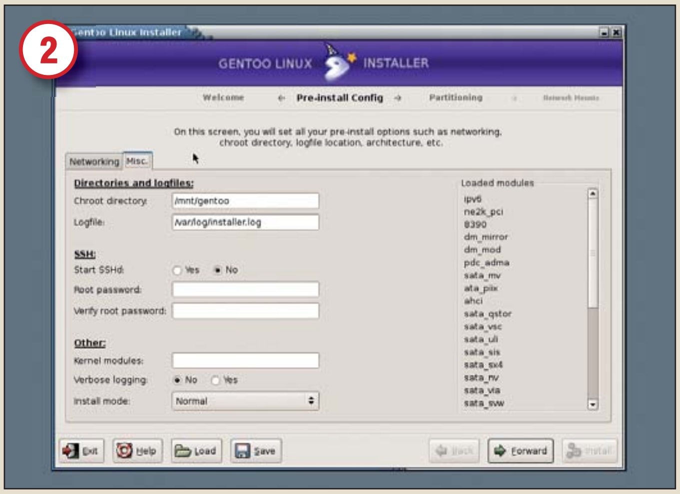
УСТАНОВКА GENTOO

1



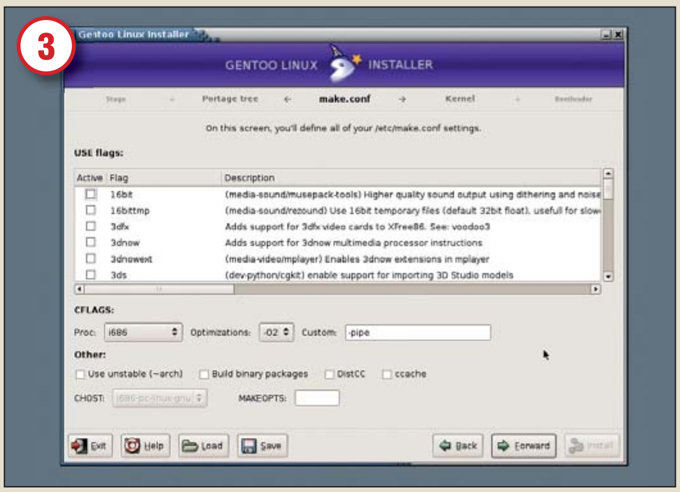
Загрузите ваш ПК с DVD и выберите Gentoo в меню загрузчика *Grub* (либо Gentoo NoFB, если у вас проблемы с видеокартой). Начнётся загрузка Gentoo Live, по окончании которой вы автоматически войдете в систему как пользователь 'gentoo' и окажетесь в рабочем столе Gnome. В нижнем левом углу находится иконка инсталлятора Gentoo Linux: кликните по ней два раза, чтобы запустить процедуру установки.

2



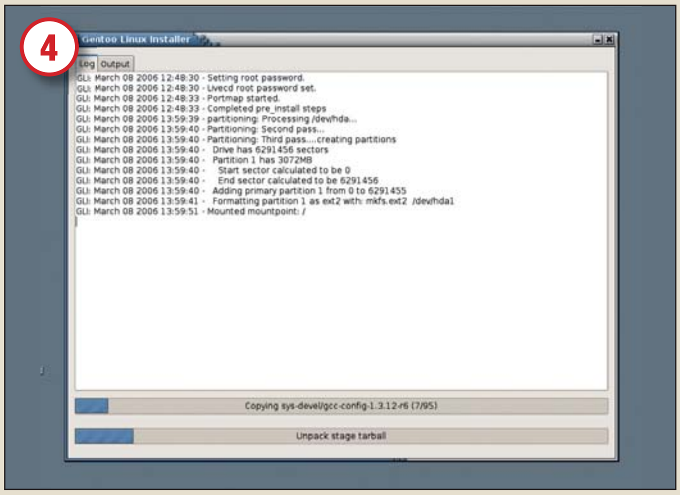
Прежде всего убедитесь в правильности сетевых настроек, чтобы инсталлятор Gentoo мог вытащить архивы исходного кода с Интернет. В каталоге **Chroot** под закладкой Misc укажите, где вы хотите собрать Gentoo, а в секции с именем Other укажите тип установки. Затем графическая утилита разбиения диска позволит вам указать место для установки.

3



Появится панель сетевых точек монтирования; оставьте значения по умолчанию и углубитесь в территорию Gentoo с его Stages и *Portage*. Если вы не хотите ждать несколько часов, пока скомпилируется каждый пакет, выберите динамичную установку Stage 3. На следующем шаге вы сможете указать USE-флаги Gentoo (настройки компиляции приложений). Если вы ещё ни разу не устанавливали Gentoo, тут-то вам и понадобится документация.

4



На этом шаге вы выбираете ядро, загрузчик и дополнительные пакеты, затем инсталлятор последний раз позволит вам просмотреть ваши настройки перед началом установки. Появится окно журнала, поясняющее, что происходит – мы предупреждали, процедура сборки из исходного кода может занять несколько часов. На фоне процесса установки вы можете запускать другие приложения с Live-дистрибутива. Удачи!

>>

<<

ИГРЫ

ALIEN ARENA 2006 GOLD EDITION



***Alien Arena* – супер-бластер, хотя на нашем диске есть игры и поинтеллектуальнее …**

Фанаты стрелялок от первого лица, на старт! Диск этого месяца содержит выдающийся трёхмерный deathmatch, *Alien Arena 2006 Gold Edition*. Как говорят сами разработчики, игра дает ответ на вопрос: «Что будет, если взять *Quake 3 Arena*, *UT*, пяток марсиан, диковатых роботов и орудий, сунуть их в соковыжималку и нажать пуск?» Игра построена на базе движка CRX (тот, в свою очередь, основан на исходном коде *Quake 2*), так что в игре нет новейших трёхмерных спецэффектов; однако фантастические изображения в ней просто супер.

Вам понадобится ПК с процессором не ниже 1 ГГц, 256 МБ памяти и графическая карта с 3D-ускорителем. Если у вас именно такая система, советуем подкорректировать настройки видео, чтобы помочь игре запуститься (играть можно, даже имея встроенную графическую карту с чипом i810). Чтобы установить *Alien Arena,* скопируйте файл **alienarena-2006ge-x86.run** в ваш домашний каталог и запустите его – либо пометив как исполнимый в файловом менеджере, либо набрав **sh alienarena-2006ge-x86.run** в консоли. После этого возникнет графический инсталлятор, и вы сможете запускать игру из установленной директории.

Alien Arena включает восемь видов оружия и 32 уровня, на которые вы поднимаетесь в однопользовательском режиме, играя против ботов; но настоящий кайф начинается в режиме многопользовательской игры онлайн. В главном меню игры выберите пункт Multi и Join Network Server – увидите список существующих игр. Найдите сервер, где есть свободные места, и принимайтесь крушить (описания клавиш находятся в в файле **README.txt**).



РАБОЧИЙ СТОЛ

ОКОННЫЕ МЕНЕДЖЕРЫ

Собрались менять рабочий стол? Возможно, Gnome кажется вам не слишком гибким, а KDE – чересчур перегруженным. Или вы ищете что-то полегче? В разделе Рабочий стол нашего диска вы найдёте подборку лучших оконных менеджеров для *X*, включая ослепительный *Enlightenment* и крепко сбитый *Window Maker*, в духе системы Next. Если вы предпочитаете традиционный рабочий стол в стиле Windows 9x, попробуйте *IceWM*, компактный и быстрый оконный менеджер – он делает огромную работу, не путаясь у вас под ногами; а если вы переживаете прилив ностальгии, установите FVWM – один из старейших оконных менеджеров.



***Enlightenment*: оконный менеджер для гедонистов.**

У нас есть и более экзотические оконные менеджеры: интригующий изыск дизайна от *Ratpoison*, для пользователей-мышененавистников и клаволюбов, и невероятно минималистичный LWM. Если вы не желаете ничего лишнего на своём рабочем столе, то LWM вам поможет: в нём нет иконок, нет кнопок, нет меню, нет вообще ничего. Возможно, это единственный оконный менеджер в своём роде. Автор объясняет, что хотел дистанцироваться от громоздких оконных менеджеров, которые «помогают ОС преступно оккупировать ваше дисковое пространство и аннексировать физическую память». Круто.

ДРУГИЕ ПРОГРАММЫ

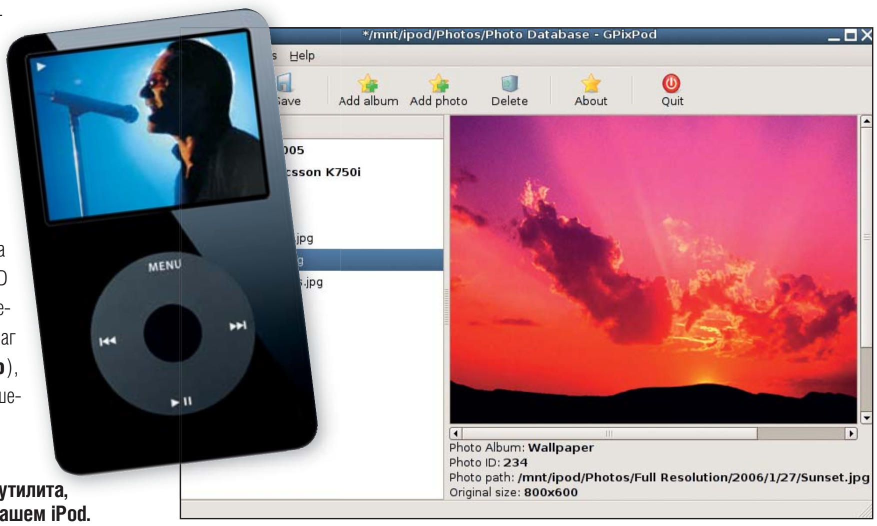
Даже записав три операционные системы на наш DVD, мы сумели найти место для релизов лучших программ, вышедших за последний месяц. Про *MythTV* много распространяться не будем, можете прочитать полный обзор версии 0.19 на странице 25; как всегда, в разделе диска **Журнал** вы найдёте полную подборку программ из обзора HotPicks.

Для разработчиков, одним из наиболее впечатляющих релизов должен стать *GCC 4.1.0* – компилятор GNU, основа инструментария для программистов почти в каждом дистрибутиве Linux. *GCC 4.1.0* привнёс несколько дополнительных оптимизаций, так что если вы регулярно собираете программы из исходных текстов и для вас важна их производительность, присмотритесь к этому релизу.

А если вы программируете на Java, попробуйте новую версию *GNU Classpath 0.90* – набор свободных библиотек классов Java для виртуальных машин и компиляторов. В данный релиз вошла реализация Graphics2D на основе Cairo (во время сборки укажите флаг **--enable-gtk-cairo**), плюс некоторые улучшения для *Swing* и AWT. Кроме того, к проекту добавлен *GNU Crypto*, поддерживающий алгоритмы шифрования и SSL.

Фронт рабочих столов подкрепился *Gossip* от Immendo – это клиент обмена мгновенными сообщениями под Gnome на базе протокола Jabber. Он разработан с Gnome'овской простотой, имеет ясный-понятный графический интерфейс, и хотя покаместь не может тягаться с *Gaim*, прогрессирует хорошо. Владельцы iPod должны попробовать *GPixPod* – *GTK*-утилиту, написанную на Python для аранжировки изображений на моделях iPod, оснащённых фотокамерой: пока в стадии развития, но уже работает, и имеет отличные перспективы.

Мы не обидели и пользователей KDE: в раздел **Интернет** помещена утилита управления блогом *Catkin-QTM*, использующая для опубликования записей системы *Blogger*, *MetaWeblog* и *Movable Type*. Это переписанная ранняя версия утилиты *Catkin*, поддерживавшей только блоги *Bloxsom*. Ну и, наконец, *KBeagleBar* предоставляет апплет панели KDE для работы с популярным поисковиком *Beagle* (построен на базе *libbeagle*; см. раздел **Dependencies** в каталоге *KBeagleBar* на диске). **LXF**



***GPixPod* – способная утилита, работает с фото на вашем iPod.**

IPOD IMAGE COURTESY: APPLE

Диск LinuxFormat

СОДЕРЖИМОЕ DVD-ДИСКА

Журнал

Article index	Список предыдущих статей LXF.
HotPicks	Последние версии программ с открытым кодом.

Рабочий стол

AfterStep	Оконный менеджер в стиле Next.
CTWM	Оконный менеджер с классическими закладками.
Enlightenment	Набитый графикой оконный менеджер.
F-Spot	Набор программ для работы с фотографиями.
Fluxbox	«Тонкий» оконный менеджер.
FVWM	Классический оконный менеджер.
GPixPod	Органайзер фотографий iPod.
IceWM	Оконный менеджер в духе Win9x.
KBeagleBar	Апплет поиска для KDE.
LWM	Ультра-минималистский оконный менеджер.
MythTV	Утилита для записи видео.
PAWM	Простой оконный менеджер.
Ratpoison	Оконный менеджер, управляемый с клавиатуры.
Tomboy	Записная книжка для рабочего стола.
Window Maker	Ещё один оконный менеджер в стиле Next.

Разработка

Classpath	Основные библиотеки классов Java.
GCC	Коллекция компиляторов GNU.
MonoDevelop	Среда разработки для Mono.
PyPE	Редактор для программистов.
Ultimate++	Инструментарий разработчика.
Visual REGEXP	Для создания регулярных выражений.

Дистрибутивы

Belenix OpenSolaris Live	
Gentoo	Дистрибутив для опытных пользователей.
SimplyMepis	Дружелюбный дистрибутив для начинающих.

Игры

Alien Arena	Трёхмерная многопользовательская стрелялка.
Mednafen	Эмулятор игровых консолей.
Sparklet	Космическая стрелялка.
Triplex Invaders	Игра в стиле Alien Invaders.
XPilotNG	Двумерные космические войны.

Графика

AutoQ3D	Инструмент 3D-моделирования.
Dia	Программа для рисования.
Glips	Редактор SVG.
Inkscape	Редактор векторной графики.
Karbon14	Векторный редактор для KDE.
Skencil	Редактор векторной графики.
SVG Test Suite	Тестовые файлы SVG.
Xfig	Программа для рисования.

Справка

LDP	Проект документации Linux.

Интернет

Blam	RSS-агрегатор.
Catkin QTM	Управление блогом.
Gossip	Клиент обмена мгновенными сообщениями.
KLinkStatus	Утилита проверки web-ссылок для KDE.
Kvpnc	Графический интерфейс к VPN-клиентам.
Thunderbird	Клиент электронной почты и новостей.

Офис

Lyx	Обработка документов.
OpenOffice.org	Офисный пакет.
Pstoedit	Конвертер PostScript/PDF.
WPClipart	Открытая галерея изображений.

Сервер

And-HTTPD	Супер-безопасный web-сервер.
Flat Gallery	Галерея web-изображений.
Schema Crawler	Просмотрщик схем баз данных.
Simba	Утилита зеркалирования.

Звук

Ardour	Работа с цифровым звуком.
Audacity	Звуковой редактор.
Banshee	Музыкальный проигрыватель.
FUPlayer	Музыкальный проигрыватель.
Jaaa	Анализатор спектра Jack.
Jack	Сервер аудио.
Jamin	Редактирование звука.
LADSPA	Система звуковых эффектов.
Muine	Музыкальный проигрыватель Gnome.
Postfish	Фильтр аудио.
Qjackctl	Qt-интерфейс для Jack.
Wavbreaker	Нарезка WAV файлов.

Система

Band Saw	Syslog-монитор для Gnome.
Hdparm	Настройка параметров жёстких дисков.
Ketchup	Менеджер дерева исходного кода ядра.
Kiosktool	Утилита киоска KDE.
Mono	Реализация .NET.
OpenKiosk	Система Kiosk.
Pessulus	Редактор для Gnome.
Qps	Визуальный менеджер процессов системы.

Главное

Allegro	Библиотека для разработки игр.
Avifile	Библиотека для чтения AVI-файлов.
CheckInstall	Создание бинарных пакетов.
CSV	Индекс содержимого диска.
GLib	Низкоуровневая библиотека.
Glibc	Библиотека GNU C.
GTK	Инструментарий разработки интерфейса пользователя.
Guile	Встраиваемая реализация языка scheme.
Kernel	Исходный код и заплатки.
libsigc	Библиотека обратных вызовов с проверкой типов для C++.
LibXML	Анализатор XML для C и инструментарий.
Ncurses	Инструментарий создания окон в текстовом режиме.
Rawrite	Запись образов дисков на дискеты.
SDL	Мультимедийная библиотека.
Smart Boot Manager	Системонезависимый менеджер загрузки.
WXWidgets	Создание графических интерфейсов на C++.



FVWM не потерял ни грамма привлекательности.

Линуксцентр представляет авторизованные курсы Mandriva Linux.

Курсы не привязаны к конкретному дистрибутиву и сертифицированы международной организацией LPI - Linux Professional Institute (www.lpi.org).

★ LNX70: Введение в Linux.

Предназначен для пользователей, которые хотят приобрести навыки использования Linux. В рамках курса рассматриваются вопросы миграции на Linux и даются базовые знания для пользователей.

Цель курса: Обучить пользователя оптимальной работе с Linux.
Требования к соискателям: Базовая компьютерная подготовка.

★ LPI101: Администрирование Linux для начинающих.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования Linux. Рассматриваются следующие вопросы: конфигурирование системы, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI101.

Требования к соискателям: Знания по Linux в объеме курса LNX70.

★ LPI102: Сетевое администрирование Linux.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования сетей на базе Linux. Рассматриваются следующие вопросы: конфигурирование периферийных устройств, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора к управлению сетями на базе Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI102.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

★ LPI201: Администрирование Linux для специалистов.

Курс предназначен для системных администраторов, которые хотят повысить свою квалификацию в области управления сложными Linux-системами. Рассматриваются следующие вопросы: настройка ядра Linux, восстановление системы, восстановление файловой системы, настройка RAID, сопровождение системы, автоматизация задач администрирования.

Цель курса: Подготовить системного администратора к управлению сложными Linux-системами в соответствии с требованиями к соискателям на сдачу экзамена LPI201.

Требования к соискателям: Знания по Linux в объеме курса LPI101 или LPI102.

★ LPI202: Сетевое администрирование Linux для специалистов.

Курс предназначен для опытных системных администраторов, которые хотят усовершенствовать навыки управлениями сетевыми сервисами на базе Linux. Рассматриваются следующие вопросы: углубленное конфигурирование сети, настройка служб электронной почты и новостей, расширенная настройка DNS, обслуживание интернет-сервера, настройка DHCP, NIS, LDAP и PAM, управление безопасностью сетевых служб, контроль за трафиком.

Цель курса: Подготовить системного администратора к управлению сложными сетевыми сервисами в соответствии с требованиями к соискателям на сдачу экзамена LPI202.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

Узнайте подробности и закажите курсы на сайтах - www.linuxcenter.ru и www.unixedu.ru

Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Содержание LINUX FORMAT DVD Май 2006

Сторона 1

Рабочий стол
AfterStep -- классический оконный менеджер
CTWM -- оконный менеджер для минималистов
Enlightenment -- оконный менеджер для ценителей прекрасного
F-Spot -- каталогизатор фотографий
FVWM -- отлично настраиваемый оконный менеджер
Fluxbox -- форк оконного менеджера Blackbox
GPixPod -- каталогизатор изображений для iPod
IceWM -- легкий оконный менеджер
KBeagleBar -- апплет для поиска в KDE
LVM -- оконный менеджер
MythTV -- средство для просмотра и записи ТВ
PAWM -- простой оконный менеджер
RatPoison -- оконный менеджер для мышененавистников
Tomboy -- программа для создания заметок на рабочем столе
Window Maker -- оконный менеджер для любителей *Step

Разработка
Classpath -- свободная реализация классов Java
GCC -- коллекция компиляторов GNU
Monodevelop -- среда разработки для Mono
PyPE -- текстовый редактор для пайтонистов
Ultimate -- набор эффективных библиотек для C++
Visual Regexp -- средство для визуального создания регулярных выражений

Дистрибутивы
BeleniX -- Unix-дистрибутив на базе ядра OpenSolaris
Gentoo -- метадистрибутив
SimplyMEPIS -- простой в использовании настольный дистрибутив

Игры
Alien Arena -- популярный шутер на мощном движке
Mednafen -- эмулятор игровых консолей
Sparklet -- космический шутер
Triplex Invaders -- клон Galaga
XPilot NG -- классическая многопользовательская игра

Графика
AutoQ3D -- программа для редактирования 3D-моделей
Dia -- инструмент для создания диаграмм
Glips -- векторный редактор
Inkscape -- продвинутый векторный редактор
Karbon -- векторный редактор для KDE
Scencil -- векторный редактор с поддержкой Python
Xfig -- редактор fig-файлов

Интернет
Blam -- программа для чтения RSS-лент
Catkin-QTM -- средство управления блогом
Gossip -- Jabber-клиент для GNOME
KLinkStatus -- утилита проверки ссылок
KVpnc -- KDE-интерфейс для настройки VPN
Thunderbird -- популярный почтовый клиент

Офис
Lyx -- программа для подготовки и верстки документов
OpenOffice.org -- открытый офисный пакет
WPClipart -- свободная библиотека клипартов
pstoedit -- конвертер ps-файлов

Сервер
And-HTTPD -- сверхнадежный веб-сервер
FlatGallery -- генератор галерей
SchemaCrawler -- генератор схем
Simba -- инструмент зеркалирования

Звук
Ardour -- аудиоредактор
Audacity -- средство для быстрого редактирования аудиозаписей
Banshee -- продвинутый аудиопроигрыватель
FUPlayer -- аудиоплеер
JACK -- звуковой сервер
Jamin -- интерфейс для аудиомастеринга
Jaaa -- анализатор спектра
LADSPA -- система плагинов
Muine -- аудиопроигрыватель для GNOME
Postfish -- аудиофильтры
Qjackctl -- Qt-фронтэнд для JACK

Система
Bandsaw -- syslog-монитор для GNOME
Ketchup -- менеджер исходных кодов ядра
Kiosktool -- настройщик киоск-опций для KDE
Mono -- свободная реализация .NET
OpenKiosk -- открытая киоск-система
Pessulus -- средство для ограничения пользователей в Gnome
Qps -- визуальный менеджер процессов
hdparm -- утилита для настройки жестких дисков

Сторона 2

MOPS Linux 4.1
IBM Websphere
TeX-Live

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Кардмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-68

Поставляется вместе с журналом LINUXFORMAT номер 9(79) Май 2006

СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование cdrecord, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку «Burn» и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на «Combust!». Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не придется использовать Linux для записи дисков, подойдет любая операционная система, способная распознать привод записи дисков (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

LINUX FORMAT

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно
Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:

ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru

Переводчики
Юрий Афанасьев, Александр Бикмеев, Павел Гладков, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Салимов, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабунио, Павел Шер.
Литературные редакторы
Родион Водейко, Иван Мищенко, Елена Толстякова
Допечатная подготовка
Мария Пучкова
Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Павел Фролов, Станислав Медведев
Отпечатано в типографии "Текст",
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейтч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор
Джулиан Джефферсон (Julian Jefferson) julian.jefferson@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Операционный редактор
Ребекка Смелли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрью Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

АВТОРЫ
Джоно Бэкон (Jono Bacon), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Энди Ченнел (Andy Channelle), Кингс Кобблер (Kings Cobbler), Дэвид Хэррис (David Harris), Энди Хадсон (Andy Hudson), Евгений Балдин, Андрей Боровский, Анатолий Евдокимов, Дмитрий Кирсанов, Андрей Маркелов, Владимир Попов, Тихон Тарнавский, Игорь Тимошенко, Алексей Федорчук, Сергей Яремчук

ХУДОЖЕСТВЕННЫЙ ОТДЕЛ
Арт-ассистенты: Джон Блекшау (John Blackshaw), Симон Годдард (Simon Goddard), Дэниель Винсент (Daniel Vincent)
Фотографии: Joby Sessions, Simon Lees, Superstock, Photodisc, Photolibrary.com
Иллюстрации: Пол Бейтмен (Paul Bateman), Крис Винн (Chris Winn), Шейн Коллин (Shane Collinge)
Создание диска: Майк Сондерс (Mike Saunders)

КОНТАКТНАЯ ИНФОРМАЦИЯ
England: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email linuxformat@futurenet.co.uk
Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Email: info@linuxformat.ru
Web: www.linuxformat.ru

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO •SAN FRANCISCO •

В СЛЕДУЮЩЕМ МЕСЯЦЕ

LINUX FORMAT 5(79) МАЙ 2006

ВСЕ О ГРАФИКЕ

LXF ведет репортаж с конференции Libre Graphics. Последние новости о статусе Gimp, Inkscape, Xara – в LXF 80!



The exact contents of future issues are subject to change

Брайан Белендорф

LXF-ИНТЕРВЬЮ

Разработчик Apache – без галстука

Обзор KOffice

Встречайте Kross, альтернативу VBA от KOffice

Обновление GPL 3

Работает ли лицензирование под присмотром Комитета?